# DRAGON USER

International edition

*The independent Dragon magazine*

60p US$2.60 October 1983



Commands to tame your Dragon

Play Bust-out and Connect

Arcade games reviewed

US author interviewed

Kill those bugs

WIN

A modem and interface from CP Engineering and Cotswold Computers


MY NAME IS DIAMOND, DAN DIAMOND I'M A PRIVATE COP. I WORK THE BIG APPLE A SEETHING METROPOLIS FILLED WITH HUMAN MISERY AND CHINESE TAKEAWAYS.
NORMALLY I ONLY DO ROUTINE DIVORCE CASES BUT WHEN SHE WALKED INTO MY OFFICE I FOUND MYSELF INOLVED IN A CASE SO STRANGE THAT
IT MADE THE BIG SLEEP LOOK LIKE A CAT NAP...
NEWS
HUNT WINS GRAND PRIX
At yesterday's Monaco Grand Prix, a hunting party strayed onto the track at the climax of the race. Cars were halted as the hounds rampaged around the circuit. "The whole place has gone to the dogs," one driver was reported as saying. The race was restarted; riders and drivers battled bitterly around the course before the Hunt thundered past the finishing line to take the chequered flag (it hasn't been seen since).
PLAYER WINS OPEN
Eagle eyed spectators were privileged to see
Salamander SOFTWARE
GSH Guild of Software Houses
17 · NORFOLK ROAD · BRIGHTON · BN1 4AA · SAE FOR FULL CATALOGUE
PRIVATE DETECTIVE DISAPPEARS
Police are baffled by the disappearance of Dan Diamond. He was last seen approaching the eerie edifice known as Franklin's Tomb, but the authorities are completely unable to find any trace of him. Citizens are asked to report any information relating to his disappearance immediately. For further details, buy FRANKLINS TOMB, a new adventure game for the DRAGON 32 and 48k ORIC-1. This adventure comes complete with a 24-page illustrated Case File. £9.95 from BOOTS, SPECTRUM, COMPUTERS FOR ALL, WEBSTERS and all other purveyors
BANANA DICTATOR
2 DEAD IN EVEREST TRAGEDY
The Everest Expedition ended in tragedy yesterday as Carl and Fre plunged down a crevic to a grisly death. Har the expedition lead was quoted as sayi "Yuk". Continued on pag
COLD WAR ON XARG ESCALAT
Thousands dead in Ice Storm
Muduras the Mu

# DRAGON USER

October 1983

**Editor**
GRAHAM CUNNINGHAM
(01-734 2631)

**Assistant Editor**
ANDREW BRIGGS
(01-734 2631)

**Software Editor**
GRAHAM TAYLOR (01-734 2953)

**Editorial Secretary**
SARAH OWEN (01-734 3454)

**Advertisement Manager**
DAVID LAKE

**Advertisement Executive**
SIMON LANGSTON (01-734 2923)

**Administration**
THERESA LACY (01-734 3454)

**Managing Editor**
DUNCAN SCOT

**Publishing Director**
JENNY IRELAND

**Subscriptions**
UK £8 for 12 issues
Overseas £14 for 12 issues
ISSN 0265-0177
*Dragon User*, 12/13 Little Newport Street, London WC2R 3LD.
Published by Sunshine Books, Scot Press Ltd. 
Typesetting by Chesham Press, Chesham, Bucks. Printed by Eden Fisher (Southend) Ltd, Southend-on-Sea, Essex.
Distributed by S.M. Distribution, London SW9. 01-274 8611. Telex: 261643



**How to submit articles**

The quality of the material we can publish in *Dragon User* each month will, to a very great extend, depend on the quality of the discoveries that you can make with your Dragon. The Dragon 32 computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Every one of us who uses a Dragon will be able to discover new tricks and quirks almost every day. To help other Dragon users keep up with the speed of the development each of us must assume that we made the discovery first — that means writing it down and passing it on to others.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

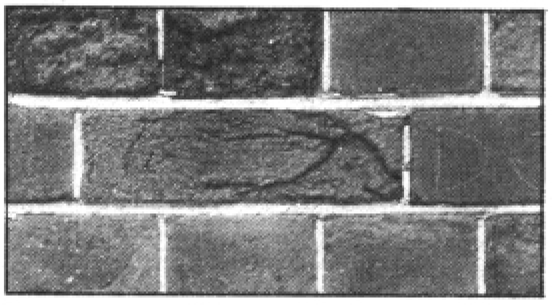
We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped, addressed envelope.

We pay for articles according to the length and the quality — it is worth making that extra bit of effort.

# Contents

SUNSHINE







# Editorial

"IT'S BUSINESS AS USUAL" was the message from Dragon Data following the announcement of a £2½ million investment package for the company at the beginning of September. Or, more specifically, "It's business as it was earlier this year and not as it was this summer" — for it is the hot summer which was held to blame for the company's financial problems. Potential purchasers headed for the beach rather than the nearest computer retailer, with the result that sales were lower than expected and Dragon Data hit problems "funding its short-term cash requirement". This shortfall between fact and forecast placed an even greater strain on the company because at the same time "it was gearing up production to meet pre-Christmas demand".

The announcement of the rescue package, reported to be in the form of loans or guarantees for loans of up to £2.5 million, also contained the news that managing director Tony Clarke was stepping down to make way for a senior executive from GEC. The entrance from the wings of GEC was not entirely unexpected. Earlier this summer the electrics giant was involved in talks with another British microcomputer manufacturer, Torch. When these talks came to nothing, it was pointed out that GEC's interest in microcomputing was likely to continue, particularly in the light of its rivals' progress in this area. With hindsight, Dragon Data's problems should not have been entirely unexpected either. The market has become increasingly competitive and other firms, such as Grundy, have suffered worse this summer while even Sinclair is said to be struggling in the US. Carving off Dragon Data from Mettoy to set it up as an independent operation was not an orthodox start, and the home computer firm may have been under-capitalised from the beginning.

New management and more money may put Dragon Data back on the right track. At any rate the company's shareholders have made their position clear — they believe that the package will ensure Dragon Data's financial stability for the foreseeable future, and are confident in the new machines it is developing. To make our position equally clear — whatever happens to Dragon Data people will still be using its machines while others produce software and peripherals for them, and we will still be producing *Dragon User*.

# Enter Dragon 64

THE EAGERLY anticipated Dragon 64 will make its first UK public appearance at this autumn's PCW show.

The Dragon 64, which is already being sold in the US, will allow Dragon Data to compete with the Commodore 64. In addition to the extra 32K of memory, the main difference between the Dragons 32 and 64 is an RS232 interface.

Dragon Data will also be launching nearly 20 new software titles at the show, ranging from arcade games and adventures to its new Forth cassette.

The games include Vikings, Monsters and Magic, Blockhead, Adventure Trilogy, Bridge and El Bandito. The company's disk drive system will also be on display during the show.

Visitors to the show will also have the chance to meet the team that produces this magazine. Sunshine, which publishes *Dragon User*, will be on stand 140. As a special discount to show visitors, we are cutting the price of our four Dragon books by £1 each — this includes our latest title *Dragon Trainer*.

The PCW show runs from the 29th September to the 2nd October at the Barbican Centre in London.

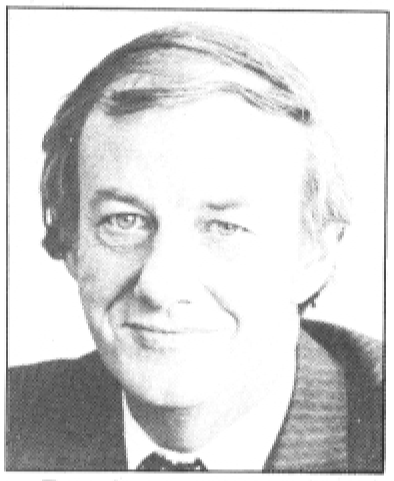
Visitors to the show can save a further 50p by taking along the admission voucher printed at the bottom of this page.

# A cash injection cures DD's 'summer sickness'

DRAGON DATA'S new chief executive is to go ahead with plans for new machines following the announcement of a £2½ million financing package for the company early in September.

Work on the next generation of Dragons, both new home computer products and small business machines, is said to be in the "late stages of development". This has encouraged the shareholders to put more money into the company.

It was also announced that Dragon Data's managing director Tony Clarke was stepping down, with a senior GEC executive coming in "on temporary secondment" to be the new chief executive.



*Tony Clarke: stepping down*

The cost of developing the new Dragons was held to be partly responsible for more investment being needed. Another factor was summer sales falling short of predictions.

However, Dragon Data argues that this seasonal slump was general and that it held on to its market share during the hot summer.

The link between a bad summer and the need for more money coloured reactions to the £2½ million announcement, so that it was interpreted as a rescue package.

But Dragon Data argues that a request for further investment is natural at this stage of its development and expects a profit for its first year of operation.

# Cumana in Dragon disk drive race

DISK DRIVE designer and distributor Cumana is following hot on the heels of Dragon Data's arrival in the disk market.

Cumana is launching its range of slimline disk drives this autumn, claiming that "this is the first time that a drive has entered the High Street with packaging to launch it into the consumer market place".

But Dragon Data has beaten Cumana to Boots and Dixons, which are stocking its drives. Priced at £275, Dragon Data's drive is a single half-height unit in a coated steel case — as featured on the front cover of July's *Dragon User*.

It has an internal power supply and can be expanded to a double disk system by inserting an additional drive. Formatted memory capacity is 184,320 bytes on a single-sided, double-density 5¼-inch mini diskette.

The disk controller can support up to four drives, either single or double-sided, and the operating system is held in Rom on the controlling card.

● The drives will be reviewed in depth in next month's issue of *Dragon User*.

# Speech software coming

A RANGE of software for use on its Speech Synthesis Module is on the way from JCB Microsystems, including a program converting text into speech.

The first cassette due for release is a Telling the Time program for young children. Most of the other programs scheduled are also for the same market — Letters, Numbers, Counting and Spelling. But JCB, whose first product was a sound extension module, is also working on a program which will convert English text into speech.

The speech module was launched in August and costs £37.95. It is based on a General Instruments speech processor which simulates speech electronically from a set of 64 individual speech sounds, known as allophones.

The speech module plugs into the Dragon's cartridge port and is operated by five new Basic commands.

JCB adds that the extra hardware supplied by the speech module produces more intelligible speech than software-based voice synthesisers.

JCB is on (0202) 423973.




# Stepping Forth on disk

FORTH HAS joined the range of languages available to Dragon disk users, courtesy of M and J Software.

The company is offering a fig-Forth implementation with an editor and microassembler for the 6809 chip on which the Dragon is based. It costs £14, which M and J claims is "cheaper than cassette-based versions produced by other companies".

M and J has also developed an upgrade for its cassette-based Forth so that buyers of Premier Microsystems' Delta drives can convert their cassette versions to disk. This upgrade costs £5.

At the moment the disk Forth is only available for Premier's drives.

M and J is on (0782) 517876.

# Dragon experts launch software

TWO OF the people involved in the birth of the Dragon have set up their own software firms — Eurosoft and Diand Software.

Andy Redman, who lined up much of the original Dragon software, is the man behind Diand, while Eurosoft is run by Gerry Quick.

Gerry was one of the people responsible for toy-maker Mettoy moving into the personal computer market with Dragon Data, which went independent last November.

The first three packages from Diand are Snakebite, Cat and Mouse plus Space Maze, and Sketch-a-puzzle. Each package costs £7.95 and Diand can be reached on (0792) 391183.

Diand's next step is to move into the educational market, where Gerry Quick also thinks that there is plenty of opportunity.

Eurosoft has a wider range of programs on offer, 13 titles in all, ranging in price from £19.95 to £7.95.

The two most expensive packages are Tape Information Management System, a database which is available on tape or disk, and Disk Doctor, which promises to ease the problems of crashed disks.

Autorun, at £9.95, should be of particular interest to Dragon users. Among other things it does away with RUN commands after loading.

Eurosoft can be contacted on (0792) 401153.




*Gerry Quick – helped conceive the Dragon, now runs Eurosoft.*




*Andy Redman – bringing his Dragon expertise to bear on Diand.*

# Seek and BCS will find

EVER HAD trouble searching for that elusive piece of equipment that was just right for your Dragon?

Well, Berkshire Computer Specialists (BCS) could have just the thing to stop you wearing out your shoes and your patience.

It is operating a subscription service for Dragon users whereby they will mail out lists of suppliers of hardware and software for the Dragon.

The entries on the list will cover the name, address and phone number of the supplier along with the type of equipment or software, a price guide, whether it is available by mail order, and possibly a comment line.

The company is offering a second year's subscription free or a £1 refund to new subscribers who provide them with details of suppliers and goods not on file.

Berkshire Computer Specialists can be reached on (0626) 833855.

# Adding to your education



*Ann Mortimer*



*Patricia Lansdowne*



*Lynn Nixon*

EDUCATION HAS generally been considered the realm of the BBC micro, but now the Dragon looks set to make inroads into the Beeb's territory.

Several people have begun producing educational software specially for the Dragon 32.

One of the leaders in this field is Ampalsoft, which produces the Cheshire Cat educational series.

Ampalsoft came into being when three Cheshire housewives decided to do something about the lack of educational programs for their children. But the company has grown in leaps and bounds in just 12 months and now has 40 people — experts in many fields — working on various projects.

The original trio — Patricia Lansdowne, Lynn Nixon and Ann Mortimer — were all Dragon owners so, naturally enough, the programs were written for the Dragon. Now, however, they are being translated for use on the BBC micro and the Oric.

The programs are not just educational — they are also entertaining. For example, a program called Super Spy combines video game skills with a history lesson on World War II.

But Ampalsoft has more than just history to offer. Among its software range are maths and science programs, and tutorials on Basic programming. Programs for 'O' Level Geography, French and German will follow soon.

Dragon users keen to learn a foreign language will be well catered for in educational software, because Dragon Data will also be marketing a series of language courses — devised by Dr Michael Gruneberg of the University College of Swansea.

The series will include courses in French, German, Spanish and Italian.

ore
fer.
are
ms,
pro-
'O'
and
arn
well
oft-
ata
ies
de-
el
sity
de
an,



# Quality rises in games market

*Aliens, insects, vultures and a karate expert – an intrepid* ***John Scriven*** *tackles them all to bring you this month's software review.*

AS THE evenings draw in and it gets too cold to lie on the beach, or even in the back garden, micro owners' thoughts turn once again to cosmic pastures. Feeling somewhat guilty that I had been neglecting arcade addicts, I have spent the last two weeks immersed not in the sea, but in a large pool of games software. (Ah, the dedication!)

As it's a year since the Dragon first appeared, one would imagine that the quality of software would have improved. This is certainly true in most cases, although one or two suspect tapes still come my way. From a purely patriotic point of view, it seems a shame that many of the programs still emanate from the other side of the Atlantic, presumably conversions of Tandy Colour Computer originals. These are often streets ahead in smooth screen movement and speed of response.

An example of this is Katerpillar Attack from Microdeal. Another Tom Mix special from the States, it is a good copy of the arcade game Caterpillar. This electronic larva appears at the top of the screen and descends a line at a time. If you shoot it, it splits up and where individual pieces have been, mushrooms appear — these need to be hit several times before they disappear. Occasionally spiders leap down on to your head, but they're not too difficult to avoid. As a copy of an arcade game, it is well written and should provide a lot of amusement particularly if you are a fan of "Gardeners' Questiontime".

In case it should appear that any bias creeps in here, I think I should point out that every time I collect a fresh pile of review software, there's always a high percentage of Microdeal cassettes. This is due solely to the enormous number of programs they produce — at the last count, it was well over 50 — so it's hardly surprising that I end up reviewing many of theirs.

Occasionally at *Dragon User*, we get sent a tape from someone new who has perhaps written only a couple of programs. If you've done this, and fail to see your cassette reviewed, then sometimes it's a more gentle way of saying that it's really not up to the general standard. The alternative would be to pull it to pieces in public. Although tastes are bound to vary, the quality has improved a great deal in the last few months, and we are quite willing to be very scathing about any software house dealing in tacky goods.

| | |
|---|---|
| Microdeal<br>41 Truro Road<br>St Austell<br>Cornwall | **Katerpillar Attack**<br>**Storm**<br>**Racer Ball**<br>**Scarfman**<br>**Dragon Invaders**<br>**Cosmic Zap**<br>**Space War**<br>**Flipper**<br>All £8.00 |
| Personal Software Services<br>452 Stoney<br>Stanton Road<br>Coventry | **Attack**<br>£7.95 |
| Programmer's Guild<br>Ahead House<br>Sandbeds Industrial Estate<br>Dewsbury Road<br>Ossett<br>West Yorks | **Ninja Warrior**<br>£8.95 |
| J Morrison<br>2 Glensdale St<br>Leeds | **Vultures**<br>£6.95 |
| Peaksoft<br>7 Hawthorn Crescent<br>Burton-on-Trent | **Deaths Head Hole**<br>**Lionheart**<br>Both £5.45 |
| JCB Microsystems<br>29 Southbourne Rd<br>Bournemouth | **Sound Extension Module**<br>£34.95 |

Anyhow, back to the games! One of the best I've seen recently is Microdeal's Storm, an adaptation of a less common arcade game. The opening display resembles a spider's web with a rectangular box at the centre. As the game starts, the web is seen to be a 3-D view of a pit, with aliens climbing the walls. You can rush madly round the top firing down at them until you run out of ammunition, or you can steer around carefully increasing your score and the level of difficulty. You choose your starting level from 1 to 15, but as the game progresses, you can apparently rise higher than 75.

The instructions are very clear and the game is different enough to warrant consideration if you have £8 to spare.

If you don't need originality, then Racer Ball or Scarfman may satisfy you. They are both versions of Pacman. The stories are the same, only the titles have been changed to protect the innocent! Scarfman has only one level of difficulty, but the joysticks seem more responsive than on Racer Ball; you can also use the keyboard if you prefer. Racer Ball has 15 levels of difficulty and has a "teleport" facility that deposits you at the top of the screen if you become trapped at the bottom.

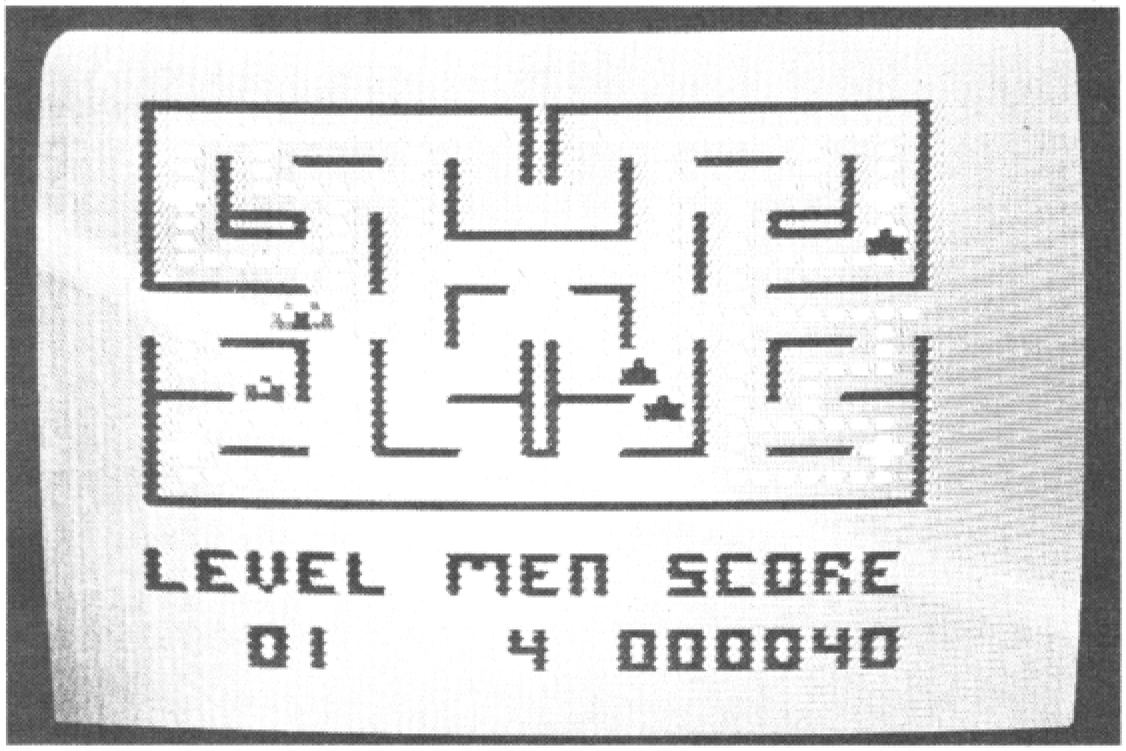
And now for something completely different. For those of you who have spent the last five years holidaying on a small planet just to the right of Alpha Centauri, I would like to announce an exciting new video game known by the highly original title of "Invaders". In this game, waves of aliens creep down the screen while you try and . . .

To relieve the yawns of boredom from the other 99.98% of the readers, however, I must refer you to the history books. Dragon Invaders from Microdeal is available complete with eight skill levels, Close Encounters theme music, flying saucers and two-player option. If you are sadistically inclined, you could give it to your granny for Christmas or sell it to a Russian tourist as an example of the latest Western secret weapon.

Those of you who are willing to boldly go, etc. etc., may find some interest in a game from J. Morrison Micros (they produce the excellent Bonka). This game is called Vultures. A group of these evil creatures starts in invaders formation at the top of the screen. They then proceed to swoop down with the one aim (and a very good aim it is too) of dropping something unpleasant on your head. Collision tactics will not work, as they take you with them as they explode. Should you manage to blast this lot to kingdom come, some harmless looking eggs hatch out to drop more galactic guano on your head. As with all games of this nature, you cannot win, and the game continues until you finally run screaming from the room. Quite a pleasant entertainment, I thought.

When faced with a pile of new software, I often divide it up into adventure, simulations, gambling games, and so on, and many arcade games get put in a pile mentally labelled "Cosmic Zap". I therefore wondered who was kidding who when I discovered this month a game purporting to be the real Cosmic Zap.

In fact, it's a reasonable program combining features from several games. ▶




*Scarfman, a version of the popular Pacman, allows a choice between joystick and keyboard control*

◀ Your task is to defend a stargate in the centre of the screen. Aliens attempt to use the stargate to get to your home star, and you have to prevent them from this form of stellar gate-crashing. A better name for the game would have been Cosmic Bouncer, though maybe this would have the wrong connotations in America, where the game originated.

After destroying several of the aliens, death satellites appear that float gently round the centre. The screen colour can be selected from a pallette of black, white or sickly green, and you can choose difficulty levels from 0 to 15. Although a simple game at level 0, at level 10 it soon becomes a test of how strong you joystick is as you try to defend your base in four directions at the same time. In spite of stealing my title, it's still quite a good game.

Attack, from Personal Software Services, puts you in the role of a prison guard on the planet Koventri. Having once been to a place with a similar name, I have every sympathy with the fleeing prisoners (Oh dear, more abusive mail from the Midlands!). Your job is to prevent the aliens from escaping. The screen display is very reminiscent of Defender, except that if you fly below the level of the hills, you will crash. Although you can see the approaching rescue craft on the upper screen, you need fast fingers before you can score highly. Although not offering all the features of a full Defender game, it is well written and very fast.

Space War is a more complicated game to learn than many of the others. Apart from having to ward off alien ships that home in on you, and a Death Star on one side of the screen that you must ultimately destroy, a black hole has positioned itself dangerously close to where you need to manoeuvre.

Using a joystick, you have to find the Achilles' heel of the Death Star where it may be successfully annihilated. The most difficult skill to acquire is steering your craft around the screen, as it appears to obey Newton's Laws of Motion far more accurately than most micro spacc vehicles. Once accelerating in one direction, it cannot be stopped unless you point it in the opposite direction and use the thrust control as retro-rockets. This game is certainly worth looking at if you are tired of run-of-the-mill space games.

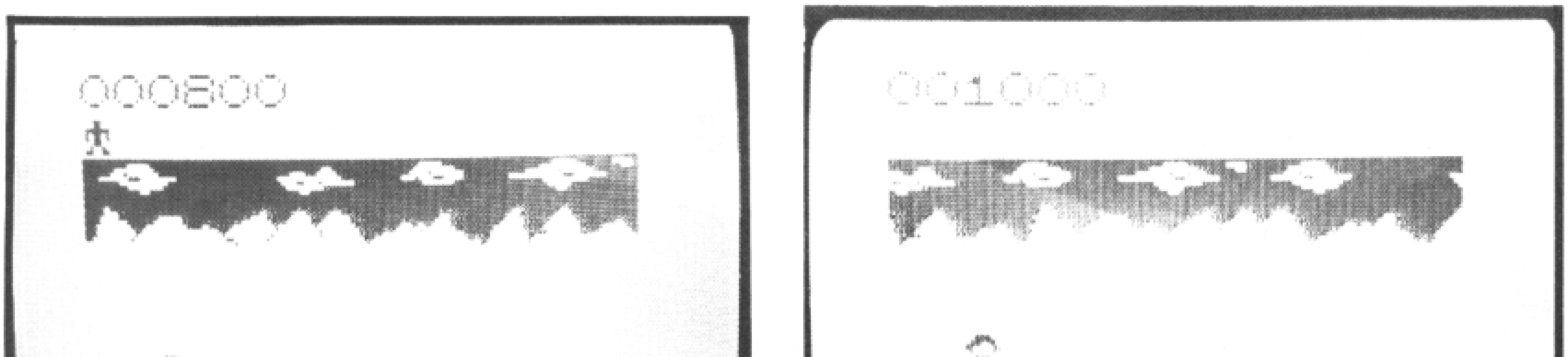
My award for the most original game of the month has to go to Ninja Warrior from Programmer's Guild. In this excellent game, you, and up to five other players, control some figure wearing a judo suit. The ground scrolls beneath his feet, and you can use a joystick to slow him down, speed him up, or make him jump in the air. The "fire" button controls a sharp karate kick that he performs.

To gain a white belt, the landscape is littered with boulders that he can jump over, but a higher score is achieved by kicking them to dust. If he survives with toe-nails intact, the tests for higher grade belts present him with fires to jump over, pits to traverse and even arrows to catch. Eventually (so they tell me!), your warrior becomes a black belt. For me, this was a highly entertaining game, using a novel idea, smooth graphics and good sound. I hope future programs from Programmer Guild will be as good.

After tackling the enormous pile of arcade action, it was pleasant to relax with a few gentler games for an evening or two.

When I was somewhat livelier at weekends, and before my wet-suit finally disintegrated, I used to enjoy wriggling through muddy passages deep under the Mendips. I was therefore interested to find amongst some adventure games one entitled Death's Head Hole. Set in the Mendips, you are one of the unlucky rescuers who have to try to extricate three lost cavers trapped somewhere inside this system. Clearly written by someone with speleological experience, all the usual fun things are present — choked passages, rock falls and sumps.

At the start of this text-only adventure (except for the map — see later), you can select your equipment from a pool of lamps, ladders, diving gear, food packs and so on, and then you start to explore the system. This contains the same main passages each time you play, although the position of the lost cavers and a few small crawls change to provide a bit of variety. A random element is provided in that shovels break, you can get hopelessly lost, or the roof can collapse on your team. If you are willing to sacrifice a few fitness points, you can look at a map of the cave, and you can even listen to see if you can hear the plaintive cries of the lost ones. Eventually, but not usually until you gain experience, you find yourself on the top of the limestone once again. ▶




*The very original Ninja Warrior from Programmer's Guild: getting a judo-suited figure successfully over boulders scores you early points*

◀ It is an adventure game with a difference and, apart from being a safe way to initiate oneself into the world of caving, it is enjoyable and challenging to play.

Lionheart, also from Peaksoft, is an adventure game in two stages. After selecting a skill level from one to five, you have to move around a forest shown on the screen by green blobs, while collecting a force of soldiers to fight for you. Preventing you from your task is an irritating bunch of Prince John's men who always seem to get in the way. If you manage to collect at least 450 men you can set sail for the Holy Land where the second half of the game takes place.

Another map appears on the screen in low resolution blocks showing villages and caves, and a river running across the middle. Saladin's castle is on the south side, and the object is to explore the land until you have amassed even more soldiers and weaponry to attack the Saracens.

Although fun to play a couple of times, it does not have the strengths of Ring of Darkness from Wintersoft. It does possess some mildly humorous (?) moments, like the kindly (sic) witch who asks you maths questions — her name is Maghi Al-Tha' acha.

Flipper is the name by which Microdeal refer to Othello (I wonder if Shakespeare knows what's going on?). In this version of an oriental board game you start with two green and two red pieces in the centre of a normal 8 × 8 chess/draughts board. You can play with a human or against the computer and there is even an option to allow the computer to play itself. The next time you feel guilty about going out and leaving your poor little Dragon on its own, you now know what to do: just load Flipper and leave it to its own devices!

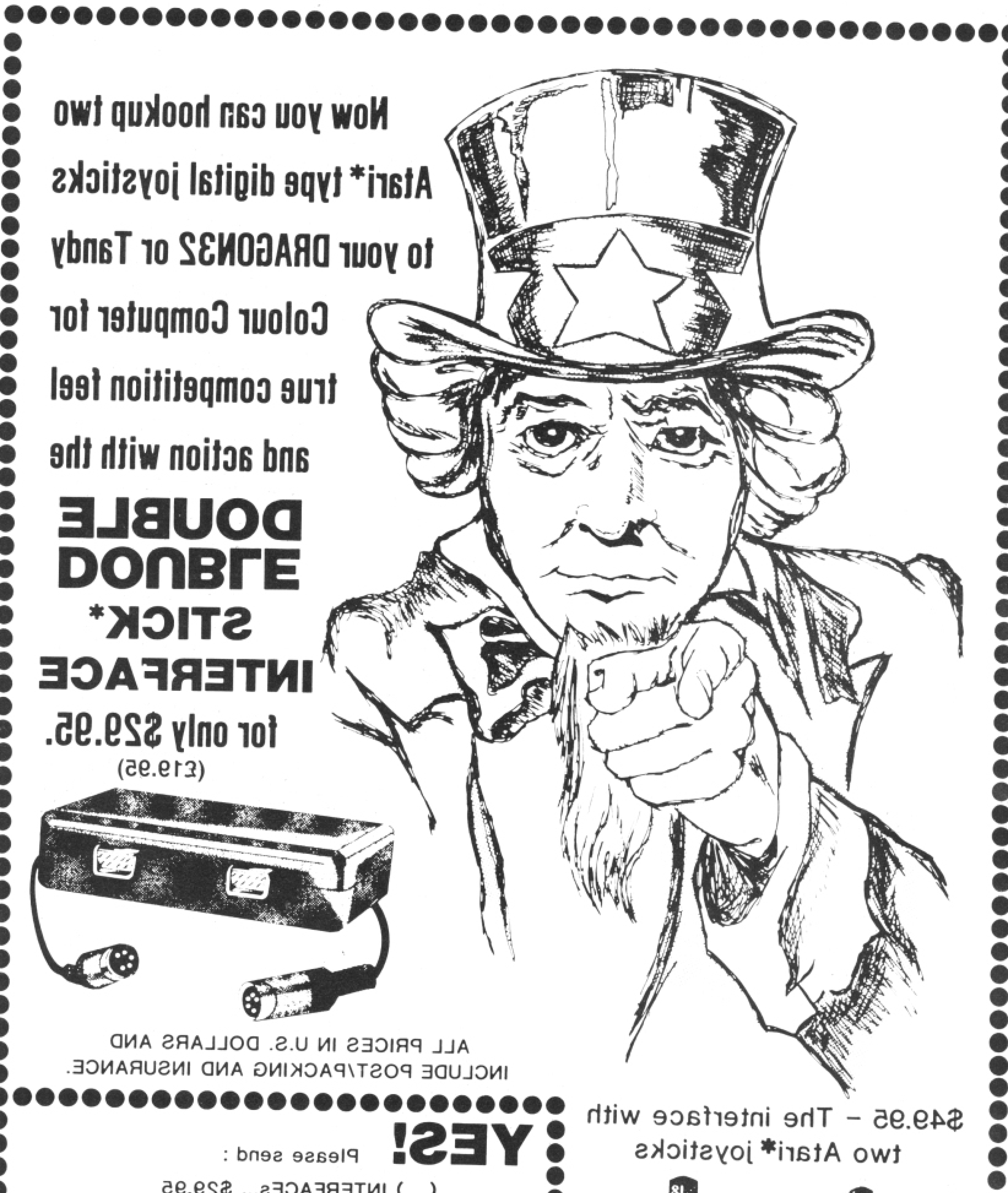
If you play against the computer, you can choose from four different levels. If you have not come across the game before, then a brief word should suffice. Taking it in turns, you place counters on the board in an attempt to have more than your opponent at the end. So that the game does not always result in a 32-32 draw, each time you trap a row of your opponent's counters between two of yours, they change colour to show they now belong to you. In the board game, the pieces have different colours on both sides and are flipped over — hence the name.

The speed of response is quite fast, but the accompanying "computer-is-thinking" type music is irritating. If you like computer board games this one's not bad. My main complaint was that there appeared little scope for cheating!

The last subject for review is not strictly speaking purely software, but more of a programming aid. When computers are designed, there seem to be many criteria that are important — cost, looks, internal structure, type of Basic, and so on. Cynics would say that the needs of the customer are always left to the last, but it would be impossible to design something that was all things to all users.

The Dragon, like all home computers, is a compromise in terms of facilities and price, and offers a fair amount of what most similar micros do. It does however have a few deficiencies in specific areas and some firms provide add-ons to make it easier to achieve certain effects. The ZX81 was a case of a machine built to a price, but with so many add-ons that it was (and still is) possible to spend a thousand pounds or more to make it have colour display, speech synthesis, discs, printers, modems and RAM in excess of 1 megabyte. Dragons are capable of supporting most of these without expensife interfacing units, but one area where they could be improved is in the case of the sound output.

One firm that helps out here is J.C.B. Microsystems. Their sound extension module plugs into the cartridge socket at the side. Measuring 12cm × 10cm × 2cm, it contains the popular AY-3-8910 sound chip (as fitted to computers such as the Oric), as well as an EPROM and a few bits of TTL. Whereas programming the AY-3-8910 is possible using hundreds of PEEKS and POKES, this sound module adds a new command MUSIC to Basic which makes the entry of tunes much easier. Not only pitch and volume are variable: the envelope can be altered to make reasonable "impersonations" of different instruments in the same way as a synthesizer achieves its effects. ■



Now you can hookup two
Atari* type digital joysticks
to your DRAGON32 or Tandy
Colour Computer for
true competition feel
and action with the
DOUBLE
STICK*
INTERFACE
for only $29.95.
(£19.95)
ALL PRICES IN U.S. DOLLARS AND
INCLUDE POST/PACKING AND INSURANCE.
$49.95 – The interface with
two Atari* joysticks
YES!
Please send :
( ) INTERFACEs... $29.95
( ) STICK/INTERFACE SETs... $49.95
I enclose a Postal Order in US funds for $............
NAME
ADDRESS
POST CODE
SPECTRUM PROJECTS
93-15 86th DRIVE • WOODHAVEN, NY 11421 USA
* Interface made by WICO. Atari is
a registered trademark of Atari, Inc.


WHEN IT COMES TO
DRAGON 32
Software
"WE'VE GOT IT TAPED"
Cuthbert in the Jungle
MICRODEAL
KEYS of the WIZARD
MICRODEAL
Cuthbert Goes Digging
MICRODEAL
INTERGALACTIC FORCE
MICRODEAL
CRAZY PAINTER
MICRODEAL
MICRODEAL

# How to become a Dragon-tamer

***Brian Lloyd** reveals some useful commands to help you train your Dragon.*

EVERY CHARACTER THAT is on the keyboard has its own code number — for example, the letter A has the code 65. It is possible to display these characters on the screen by using their code numbers.

To do this we use the CHR$ command followed by a number in brackets. This tells the computer that we want it to display the character with the code number that we have specified.

CHR$ is a function and so we have to tell it which character we want displayed. Try typing in:

PRINT CHR$ (65)

This line tells the computer to display the character which has the code 65 (the letter A) on the screen.

It is also possible to display special coloured blocks called graphics symbols by using the CHR$ command. For example, if you type: PRINT CHR$ (175) a blue square will be displayed.

Here is a short program which displays every character that the Dragon is capable of displaying on the screen and then displays the characters one by one with their codes:

```
10 CLS
20 FOR N = 32 TO 255
30 PRINT CHR$ (N);
40 NEXT
50 FOR N = 32 TO 255
60 PRINT @ 333,CHR$ (N);" = ";N
70 FOR M = 0 TO 300:NEXT M
80 NEXT N
```

Let's first look at lines 10 to 40 and see how they work.

The CLS command clears the screen and the FOR . . . NEXT loop in line 20 starts off the loop with the value of N set at 32.

Line 30 then PRINTs the character with the code number N before line 40 sends the computer back to line 20 which adds one to the value of the variable N. The program continues round and round the loop until the value of N equals 255.

The second part of the program is very similar to the first part, except that the characters are PRINTed one by one together with their codes.

Line 60 handles the PRINTing of the characters, using the PRINT@ command, and line 70 causes the delay.

A full list of the codes for the graphics symbols is on page 138 of the Dragon manual.

The program printed with this article uses the CHR$ command. It is a kind of simple Space Invaders-type game with only on invader. You move your base left and right with the left-and right-arrow keys. To fire, you press the F key. You have three lives and an infinite number of aliens.

If you read through the program you should be able to understand how most of the lines work, but you probably won't be able to understand how each line contributes to the program. For this reason I have split the program up into routines to show you what each section does:

Lines 20-40 — Set up the variables. The variable 'BASE' determines where your base is on the screen, 'SHIP' determines where the Invader is and 'LIFE' keeps a record of how many lives you have left.

## Helpful hints on using the CHR$ command, adding sound to your Dragon and editing

Lines 50-60 — Display your base. The spaces left on either side of the yellow blocks (CHR$ (159) is a yellow block) make sure that no trails are left behind the base when it moves.

Line 70 — Displays the Invader.

Line 80 — Checks to see if you are dead yet (the END command stops the program).

Lines 90-110 — Check to see if you are pressing a key and take appropriate action. CHR$ (9) is the code for the right-arrow key and CHR$ (8) is the code for the left-arrow key. Adding or subtracting one to the variable 'BASE' has the effect of moving the base right or left one square respectively.

Lines 120-130 — Make sure that your base hasn't gone off either edge of the screen.

Line 140 — Checks to see if you are pressing the 'F' key. If you are (and there isn't a missile on the screen already), it assigns the number 1 to the variable 'MISSILE' (this is so we can check to see if there is a missile on the screen) and assigns the value of the variable 'BASE' plus one to the variable 'MM' (this variable determines the position of your missile).

Line 150 — If there is a missile on the screen moves it up one line.

Line 160 — PRINTs the missile on the screen and rubs out the missile behind it.

Line 170 — Checks to see if you have hit the Invader. If you have, every different coloured block is PRINTed over the Invader in an explosion effect.

Line 180 — Checks to see if your missile has gone off the screen, deleting it if it has.

Lines 190-230 — Once in every 11 times around the program the Invader moves. This routine moves it left or right randomly.

Lines 240-250 — Make sure that the Invader doesn't go off the screen.

Lines 260-300 — This section controls the Invader's bombs using the same sort of routine as the one controlling your missiles, except that they come down instead of going up. ▶

```
10 CLS
20 BASE=431
30 SHIP=RND(32)+64
40 LIFE=3
50 PRINT@BASE," ";CHR$(159);" ";
60 PRINT@BASE+31," ";CHR$(159);CHR$(159);
   CHR$(159);" ";
70 PRINT@SHIP,"Y";
80 IF LIFE=0 THEN PRINT@233,"YOU'RE DEAD!!":END
90 A$=INKEY$
100 IF A$=CHR$(9) THEN BASE=BASE+1
110 IF A$=CHR$(8) THEN BASE=BASE-1
120 IF BASE>445 THEN BASE=445
130 IF BASE<416 THEN BASE=416
140 IF A$="F" AND MISSILE=0 THEN MISSILE=1:MM=
   BASE+1
150 IF MISSILE=1 THEN MM=MM-32
160 IF MM>0 THEN PRINT@MM,"↑";:PRINT@MM+32," ";
170 IF MM=SHIP THEN FOR N=143 TO 155 STEP 16:
   PRINT@MM,CHR$(N);:FOR N=0 TO 30:NEXT M:NEXT N:
   HITS=HITS+1:SHIP=RND(32)+64
```

Continued on page 21

```
180 IF MM<96 AND MM>0 THEN PRINT@MM," ";:MM=0:
    MISSILE=0
190 G=G+1:IF G<10 THEN GOTO 240
200 G=0
210 PRINT@SHIP," ";
220 Z=RND(2):IF Z=1 THEN SHIP=SHIP+1
230 IF Z=2 THEN SHIP=SHIP-1
240 IF SHIP<63 THEN SHIP=63
250 IF SHIP>94 THEN SHIP=93
260 IF RND(5)=1 AND BOMB=0 THEN BOMB=SHIP
270 IF BOMB>0 THEN BOMB=BOMB+32
280 IF BOMB>0 THEN PRINT@BOMB,"*";:PRINT@BOMB-32,
    " ";
290 IF BOMB=BASE+1 OR BOMB=BASE+32 OR BOMB=BASE+32
    THEN FOR N=0 TO 7:CLS(N):FOR M=0 TO 50:NEXT M:
    NEXT N:LIFE=LIFE-1:CLS
300 IF BOMB>479 THEN PRINT@BOMB," ";:BOMB=0
310 PRINT@0,"SCORE:";HITS;"   LIVES:";LIFE
320 GOTO 50
```

Line 310 — Displays your score and how many lives you have left.

Line 320 — This line goes back to line 50 to carry on with the program.

## Sound

This program can be improved with the careful addition of some SOUND.

The Dragon provides two ways of entering sound to your program. We will look at one of these — the command SOUND.

There are two instructions you must give when entering a SOUND command — the pitch and the duration. Try typing in:

SOUND 10,3 followed by ENTER, of course.

With the pitch instruction the numbers range from 1, the lowest note, to 255, the highest note; 89 is middle C on a piano. The duration also ranges from 1, the shortest duration, to 225 for the longest.

For the full range of tones type in the following little program:

FOR N = 1 TO 225:SOUND N,1:NEXT

This gives the whole range of sounds available with the SOUND command. To try the different combinations of pitch and duration type in the next program:

```
10 CLS
20 INPUT"SELECT PITCH (1 TO 255)";P
30 INPUT"SELECT DURATION (1 TO
   255)";D
40 SOUND P,D
50 GOTO 10
```

This little program, when RUN, will ask you to select the pitch, before storing that number in the variable P. It will then ask you for the duration and store the number entered in D.

Line 40 uses the variables P and D to play the note for the duration required.

Line 50 returns you to the start of the program so you may try some more combinations.

Now that you have learnt about the SOUND command you can add some sound to your Invader program, but first you must learn how to edit.

## Editing

Fortunately, your Dragon is equipped with an EDITOR to help you correct lines without re-typing them in.

Type the following program line in exactly as it is:

10 PRRNT "THEIR AR A LOTT OF MISSTAKES INN THISS LINEE!"

You can easily spot all the mistakes in the line. To use the EDITOR to correct the line you must first type in:

EDIT 10

The Dragon will print the number 10 on the screen followed by a space. Press the space bar twice and you will see the letters P and R appear with the cursor moving along in front.

The first correction to make is to change the R to an I, so press the C key (for Change) and then press I. The letter I will appear after the R with cursor in front.

**"The SOUND command can liven up your programs and make them a lot more interesting"**

Carry on pressing the space bar until the E in THEIR appears. Now type C followed by R, then C followed by E. The word THERE has now been corrected.

The next stage is to add an E after the AR, so carry on pressing space until the R of AR appears. Now type I (for Insert) followed by E.

Hold the SHIFT key down and press the up-arrow key to leave the Insert mode.

Type 6 followed by the space bar and the next six characters will appear. We now need to get rid of the extra T so press D (for Delete) and one of the Ts will be erased.

Type 6 followed by the space bar again and you will be ready to delete the extra S, again by pressing the D key.

See if you can delete the extra letters in INN, THISS and LINEE on your own using the method shown here.

When you have made all the corrections, or if you want to see how you are getting on with the corrections type L and the whole line will be displayed with the line number underneath ready for you to make any more corrections.

When you have finished with the line press the ENTER key and the corrected line will be displayed before the computer goes back to normal command mode.

Here is a full list of the EDITOR commands together with what they do:

SPACE — Move cursor along the line.
C character — Change the next character for the one specified.
n C character — Change the next n characters (where 'n' is any number) for the ones specified.
I — Insert all the following characters after the last character.
D — Delete the next character.
n D — Delete the next 'n' characters.
H — Hack (or cut) off the rest of the line and then enter Insert mode.
X — Go to the end of the line and enter Insert mode.
S character — Search for the specified character and move cursor to that position.
n S character — Search for the 'n'th occurrence of the specified character and move the cursor to that position.
K — Delete all the line from the cursor position.
n K — Delete the next 'n' characters after the cursor.
L — Display line and return to EDIT mode.
← — Move cursor backwards along the line.
n ← — Move the cursor back 'n' spaces.
SHIFT ↑ — Leave Insert or Change mode.
ENTER — Leave EDITOR mode.

Now you can change the Invader program, as follows:

```
140 IF A$ = "F" AND MISSILE = 0
    THEN MISSILE = 1:MM = BASE +
    1:SOUND 100,1
160 IF MM>0 THEN PRINT@MM
    "↑";:PRINT@MM + 32," ";:SOUND
    200,1
170 IF MM = SHIP THEN N = 143 TO
    255 STEP 16:PRINT@MM, CHR$
    (N);:SOUND N,1:FOR M = 0 TO
    30:NEXT M:NEXT N:HITS = HITS +
    1:SHIP = RND (32) + 64
280 IF BOMB>0 THEN PRINT@BOMB,
    "*";:PRINT@BOMB - 32, "
    ";:SOUND 255,1
290 IF BOMB = BASE + 1 OR BOMB =
    BASE + 32 OR BOMB = BASE + 32
    THEN FOR N = 0 TO 7:CLS
    (N):SOUND N + 1,1:FOR M = 0 TO
    50:NEXT M:NEXT N:LIFE = LIFE -
    1:CLS
```

The SOUND commands which you have added to the above lines make the following changes to the program:

Line 140 produces the sound when you fire your missiles.

Line 160 makes the noise as the missile goes up.

Line 170 produces the sound effects when the Invader is hit.

Line 280 makes the noise for the Invader's bomb.

Line 290 makes the noises when you get blown up.

The SOUND command can liven up your programs and make them a lot more interesting to use.

However, you shouldn't use too much sound as it slows the program down by an amazing amount. ■

# for the outer limit in pure explosive action

**TASKSHIP**
**£8.95**

You are in command of the leading spacecraft on a mission to quell a rebellion on the mining planet Phisox. The enemy have set up missile defences and fuel dumps on the ground, and tanker ships and defending sources in flight, guarding the access to the inhabited cavern. You must penetrate as far as possible, destroying defences, stealing fuel and avoiding meteors. Dock with a tanker and you will refuel while in contact, but beware the pull of the black holes and may the force be with you, ie. you fall in.
Joystick or keyboard option.
You can play Taskship as it stands or you can adapt or replace the scenario using the draw option.

**PACDROIDS £8.95**

Unlike any other "PAC" game you've ever seen — guide your Mk XIX Drone cruiser through the spaceways clearing a path to the power points to arm your smart-bombs — but beware the mines and the mothership.

★ Super Saucer lays destructo mines!
★ Super bomb disintegrates everything in your path — right up to the wall!
★ New maze every 10,000 points for escalating difficulty!
★ Multi-player ability — up to 4 players in competition!
★ Super high resolution graphics and machine language for ultra-speed!
★ More sound — more action — more features than any "PAC" game in existence!

**NINJA WARRIOR**
**£8.95**

New from Arcade Master Charles Forsythe! The most original game ever produced. 16 levels of breathtaking action. Up to 6 players in competition, keyboard or joystick control.
Guide your Ninja through boulders, fire, pitfalls, flaming meteors, and Ninja Masters to attain — NINJA GRANDMASTER.
A totally awesome experience in arcade action begins with smashing rocks with your bare feet and leads to the most wicked combat stages you are likely to experience — anywhere.

**EASY YOU SAY —**
**GO FOR IT, WE SAY**



**Special Offer**
Buy any 2 of the above and get a 3rd program free (Aggressor) rrp £5.

# SWORD £8

**Sword features:**

★ 16x16 page memory display.
★ Hexadecimal or screen-code byte representation.
★ Cursor memory writing in hex or keyboard characters.
★ Flagging to show occurrences of a chosen byte value.
★ Insert or delete cursor editing.
★ Cursor position available as parameter for other commands.
★ Status page shown on every command entry, including:
  ★ Address last under cursor (M).
  ★ Mnemonic and binary representation of byte at cursor.
  ★ Flag Search Value (S).
  ★ Label Address (L) — (destination for jump calculator).
  ★ Register display with condition code analysis.
  ★ 8 breakpoints with contents, address and status.
★ Cursor controlled register writing.
★ Cursor selection of 8 breakpoints.
★ Breakpoints can be positioned, activated and de-activated.
★ Flag to show which breakpoint caused last exit.
★ Hold command to freeze screen after exit.
★ Execution from typed-in address or cursor position.
★ Block copy (move), also serving as Fill facility.
★ Binary and Mnemonic code entry.
★ Long and short relative displacement calculator.
★ SWORD program is relocatable.
★ Some Assembler and Disassembler facilities.

*Prices shown include p&p.*

PG

*Cheques/POs to:*

Programmer's Guild (UK)
Ahed House
Sandbeds Industrial Estate
Dewsbury Road
Ossett, West Yorkshire

## Order with confidence

All programs in this advert carry a 10-day money-back warranty — if you are disappointed in the programs send them back for a full refund!!!

# Now meet 'Son of The King'

*An exclusive interview with a US designer of top-selling software from* ***Craig Hanna***

YOU ARE sitting at home looking at your monitor, your Dragon in front of you. On screen you see a Junior-size gorilla climbing vines, chains, and trying to rescue his big King of a dad from the villain's evil clutches.

No, you're not dreaming. Junior's Revenge will be available in October for the Dragon system. The much-awaited sequel to the King, Junior's Revenge is being marketed in the United States through Computerware in California.

Computerware's owner, Paul Searby, told me that negotiations were under way for a UK distributor — probably either Microdeal or Dragon Data. He stressed that Junior's Revenge would be available in October, ready for the Christmas market.

Paul arranged a meeting with BJ, the programmer of Junior's Revenge. In the 2½ years BJ has been writing for the Dragon and the Colour Computer, this is the first interview he has granted, and he preferred not to reveal his full name.

BJ is the author of many Dragon programs. He wrote MagiCube, Ghost Attack, Storm, Rail Runner, Blockhead (a Q-Bert game), Colour Connection, Home Money Manager, and many others.

With a background in drafting and a love of art, he began fooling around with computers in 1977. He started with a Southwest Technical computer — quite different to a Dragon 32.

BJ began working for Computerware about 4½ years ago and has been programming 6809 for 2½ of those years. He wrote MagiCube, a computer version of Rubic's Cube, as his first "real" 6809 game for the Colour Computer, and hasn't stopped since.

One of his favourite pastimes, appropriately enough, is going to video arcades. "My favourite arcade games are probably Donkey Kong Jr and Tempest," he told me. He likes the idea of Dragon's Lair, a new interactive laser disc video game, but said he hasn't had the chance to play it because everyone's been lined up for it when he goes to the arcade.

Of the games he has written himself, Junior's Revenge is clearly his favourite.

"There are a little over 10 screens in the whole game," BJ explained. "Some of them are the intro screens, some between demo and play, and there's a screen that displays your four highest scores. There are four (game) screens: the vine screen, the chain screen, the jump board, and Luigi's hideout." Luigi is the name of the Mario character in BJ's game.

It took BJ about two months to complete Revenge. "It took more time to try to figure out the size of Junior, the fruit, Luigi, the Vinegators, and where everything should be," BJ said.

"The hardest part was just trying to get Junior to climb up a vine, go running across, and have him fall. Next to that would have to be just trying to create the screens, and trying to put that many screens into a program.

"Coming up with the screen that says 'Junior's Revenge'," BJ said jokingly, "was the easiest part."

Since working with the Dragon to produce UK versions of his games, BJ has become quite a fan of the British machine.

"I really like the Dragon," he said. "For the price, you're getting a computer that has features better than those of the Colour Computer. I like the keyboard. The Dragon has a very nice feel — it's kind of like you're working on a real typewriter." Sometimes BJ worked for many hours at a time on Junior's Revenge, and a true keyboard was helpful.

BJ even went so far as to write the protection for Revenge. He told me: "The only thing I can say (about the protection) is that it'll give 'em some problems."

As a programmer who relies on selling software for his livelihood, BJ frowns upon software pirating. He hates the idea of unprotected versions of his program being marketed without a single cent being paid and hopes his protection will be good enough to withstand assault by the toughest pirate.

BJ continued to describe some of the intricate details he added that make Junior's Revenge such an outstanding game. "When you first push the fire button and release it you'll see a platform above you with a little chain hoist and Luigi is up there raising up father's cage. Junior is walking across the bottom and gets to this big gaping hole and he's wondering, 'What's going on, my father was here!' "

Such details improve any game. Entertaining "intermissions" between screens can give a player an interesting glance at something new or just time enough to wipe sweaty palms and get ready for the next exciting screen. Once I picked up my joystick and began playing Junior, I had a hard time putting it down.

"I was impressed by how much I could actually squeeze in (to memory)," BJ said. "This game takes up just about every free byte. It doesn't go into the Basic, but just about everything that's left."

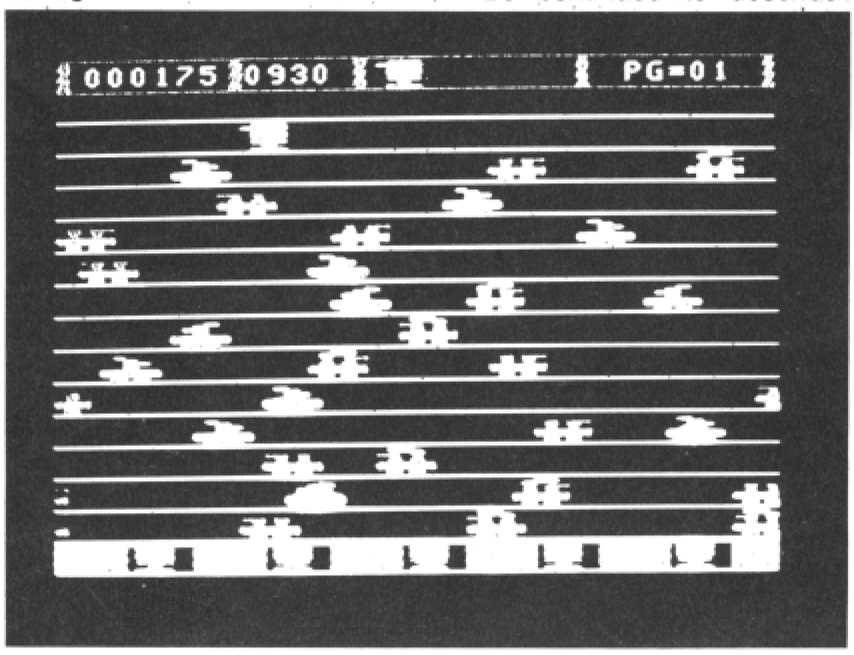
When asked how he felt Junior's Revenge compares to the arcade version of Donkey Kong Jr, BJ said: "There's certain play action that's not the same, but I tried to figure out what was lacking in Donkey Kong Jr or what would've been nice if they had put it in; then I put it in Junior's Revenge."

BJ is thinking about writing other Donkey Kong spin-offs in the future, but doesn't have any definite plans yet. He said he's taking a vacation from programming.

He has some advice for people interested in programming: "If you're trying something big (in regards to memory size), don't just sit down and write the first line of code and try to write it in one sitting. Try to break it up into little modules and stretches (of programming). Test your modules or subroutines. Make sure they're free of any errors.

"Basically, submit your programs to someone like — in the US — Paul Searby at Computerware or Ron Krebs at Mark Data. These companies are always looking for good software.

"Go to the arcades and see what's happening, subscribe to the (computer) magazines that are available." Sound advice from a successful programmer. ■




*Rail Runner is just one of the programs designed by BJ*

# BUST-OUT!

***G. Singh** challenges you to test your aim and reflexes and breach the wall.*

BUST-OUT IS an adapted version of the popular game of Breakout. In Bust-out, the object of the game is to knock out as many of the bricks in the wall as possible with the three chances which you are given at the start of a game. The bat is controlled directly from the keyboard using the left and right arrow keys.

Bust-out utilises three machine-code routines: the first to move the bat, the second to check whether all the bricks have been cleared and the third to set up the wall at the start of each game. These routines are called up in the program by EXEC31000, EXEC32050 and EXEC31200 respectively. There is also a facility to enter your initials when you have achieved a high score, and a bonus chance is given if you manage to score more than 3,000 points. An added element to make the game more challenging is that every time a sheet is cleared the length of the bat is decreased by one.

In line 10 the machine-code routines are initialised and the Dragon is switched to its faster mode of operation. In line 20, Z$ and BL$ are blanking strings used to blank out parts of the screen. In lines 30-40, the variables are set to their initial values. Line 50 asks what level you would like to play at. Lines 60-70 check that the level, L, chosen is valid. The value of L determines the speed of the bat. A level of 5 gives the highest speed of bat and 1 gives the lowest. Line 80 sets up the wall.

Line 100 initialises the variables controlling the movement of the ball. At the beginning of each game the ball can start anywhere on the 11th line on the screen but always moving towards the left. Line 110 prints the current score and the high score together with the initials of the holder of the high score. Line 120 checks whether the ball has hit the sides of the screen, and if so, reverses the horizontal direction of the ball.

Line 140 checks whether the next position to be occupied by the ball is a brick or not. If it is, the brick is removed and the horizontal direction of the ball is reversed. Also the score is increased according to the position of the brick. Thus bricks at the

```
1 '*** BUST-OUT ***
2 '*** BY G. SINGH ** 1983 ***
10 CLS0:GOSUB1000:GOSUB2000:POKE&HFFD7,0:CLEAR237,31000
20 Z$=STRING$(32,143):BL$=STRING$(11,128)
30 SC=0:HS=0:N$="":LB=5:POKE32100,5:POKE32101,210:POKE32102,5:POKE32103,217+5-LB
:POKE32104,LB
40 F=0:LI=3:FORI=1TO255STEP10:SOUNDI,1:NEXT:FORI=255TO1STEP-10:SOUNDI,1:NEXT
50 PRINT@6,"LEVEL (1-5)?";:L$=INKEY$:IFL$=""THEN50
60 IFL$>"5"OR L$<"1"THEN50ELSE L=VAL(L$)
70 PRINTL;:FORI=1TO200:NEXT:CLS0
80 EXEC31200
90 FORR=23TO0STEP-1:PRINT@R,"bustout";CHR$(128);:SOUNDR*10+1,1:NEXT:PRINT@0,Z$;
95 '*** BALL MOVEMENT ***
100 XD=-1:YD=-1:Y1=11:X1=RND(31):Y2=Y1
110 PRINT@0,"SCORE";SC;:PRINT@17,"HIGH";HS;N$;
120 X2=X1+XD:IFX2>31OR X2<0 THEN XD=-XD:SOUND200,1:GOTO120
130 IFL>2THEN EXEC31000
140 IFPEEK(1024+32*Y2+X2)>=144THEN POKE1024+32*Y2+X2,128:XD=-XD:SC=SC+10-Y2:SOUN
D150,1:GOTO120
150 IFL>3THEN EXEC31000
160 Y2=Y1+YD:IFY2>15THEN LI=LI-1:IFLI>0THENGOSUB500ELSE IF(SC>=3000AND F=0)THEN
F=1:GOSUB710:GOTO80ELSEGOSUB600:GOTO40
180 IFL>1THENEXEC31000
190 IFY2<1THEN YD=-YD:SOUND200,1:GOTO160
200 IFL>4THENEXEC31000
210 IFPEEK(1024+32*Y2+X2)>=144THEN POKE1024+32*Y2+X2,128:YD=-YD:SC=SC+10-Y2:SOUN
D150,1:GOTO160
220 IFPEEK(1024+32*Y2+X2)=131THEN YD=-YD:SOUND175,1:GOTO160
230 POKE1024+32*Y2+X2,48:POKE1024+32*Y1+X1,128:X1=X2:Y1=Y2
240 EXEC32050
250 IFPEEK(32116)=0THENGOSUB700:GOTO80
260 EXEC31000:GOTO110
490 '*** IF BALL IS MISSED ***
500 POKE1024+32*Y1+X1,128:PLAY"T20V30CGCGCGCGCGCGCG"
510 X1=RND(31):Y1=9:Y2=9:XD=-1:YD=-1
520 SC=SC-INT((SC*L-5)/10):PRINT@0,"SCORE";SC;:RETURN
```

back of the wall are worth more points than those at the front. Line 160 checks if the ball has been missed, and if it has, the number of chances is decreased by one. If all the chances have been used up then the program goes to the End of Game subroutine. If, however, you have scored 3,000 points or more, then you get a bonus chance.

In line 190, if the ball has reached the top of the screen, then its vertical direction is reversed. Line 210 checks whether the next position to be occupied by the ball is a brick. If so, the brick is removed and the vertical direction of the ball is reversed. The score is increased accordingly. Line 220 checks if the ball has hit the bat and reverses the ball's direction if it has. Line 230 moves the ball to its new position and lines 240-250 check if all the bricks have been cleared. Location 32116 contains a 0 when the bricks have been cleared. Lines 130, 150, 180 and 200 control the speed of the bat depending on the value of L.

When the ball is missed control is transferred to the subroutine at line 500. Line 500 resets the variables controlling the ball and in line 520 a penalty is paid for missing the ball. The score is decremented. The amount by which the score is decremented depends on which level you are playing at.

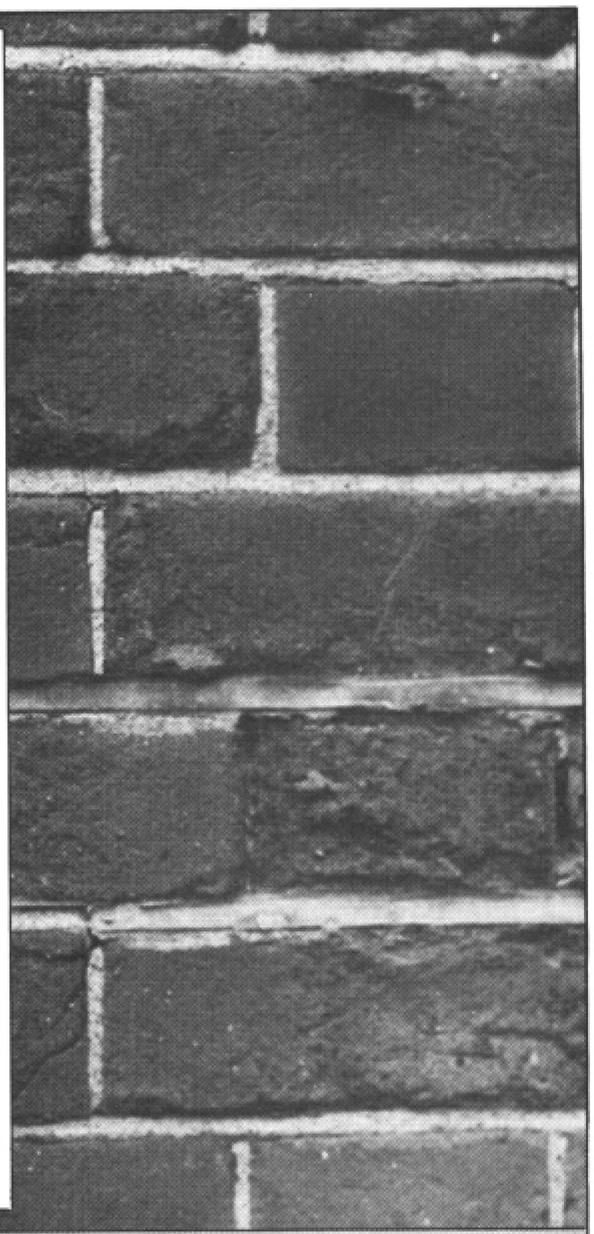
When all the chances have been used up the program goes to the subroutine at line 600. Line 600 checks if you have scored a new high score, and if so goes to the high score subroutine. Lines 610-640 tell you that the game has ended and asks you to press any key, except the left and right arrow keys (and of course, the break key), for another game. When the key has been pressed, the score is reset to 0, the length of the bat is reset to 5, and the relevant parameters in the machine-code routines are initialised.

When all the bricks have been cleared control is transferred to the subroutine at line 700. Line 700 decreases the length of the bat by one and line 720 initialises the parameters in the machine-code routines.

At the end of a game, if you have achieved a high score the program goes to the high score subroutine at line 800. Line 810 allows you to enter up to three of your initials, so that you may be identified with the high score.

## Program Variables

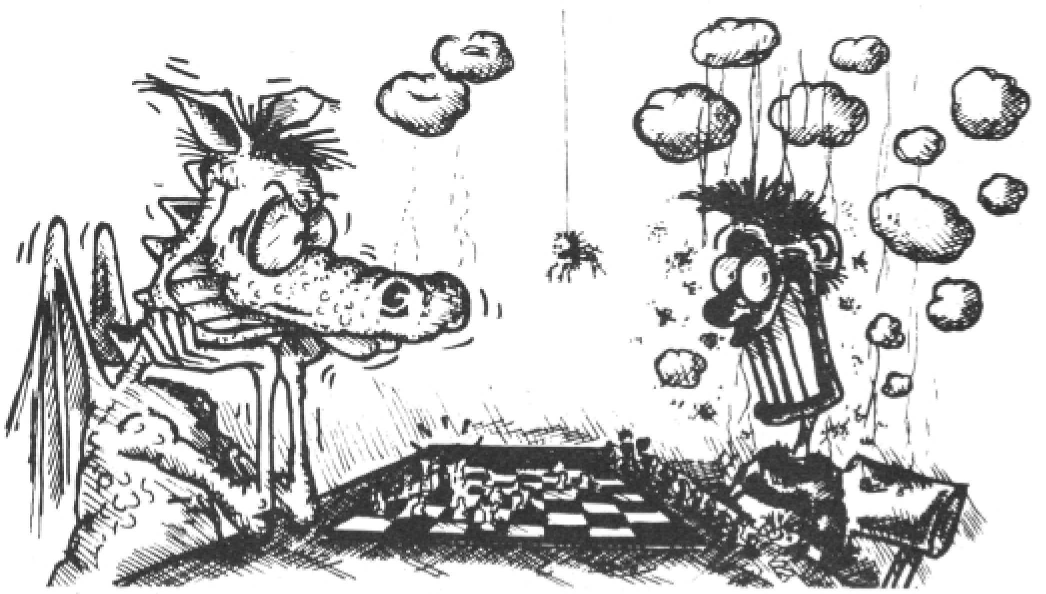
SC = score
N$ = name of holder of high score
HS = high score
LI = no. of chances
L = level of play
LB = length of bat
XD = horizontal direction of movement of ball
YD = vertical direction of movement of ball
X1,Y1 = last position of ball
X2,Y2 = position of ball
F = flags indicate that bonus has been awarded



```
590 '*** END OF GAME ***
600 IF SC>HS THENGOSUB800
610 SOUND100,1:PRINT@362,BL$;:FORT=1TO200:NEXT:PRINT@362,"END OF GAME";
620 PRINT@417,"HIT ANY KEY FOR ANOTHER GAME";:AN$=INKEY$:IFAN$=""OR AN$=CHR$(8)
OR AN$=CHR$(9)THEN610
630 PLAY"T20V3001CDEFGABBAGFEDC;02CDEFGABBAGFEDC;03CDEFGABBAGFEDC;04CDEFGABBAGFE
DC"
640 CLS0:SC=0:LB=5:POKE32103,217+5-LB:POKE32104,LB:RETURN
690 '*** IF SCREEN CLEARED ***
700 LB=LB-1:IFLB<1THENLB=1
710 CLS:PRINT@458,"WELL DONE":FORI=1TO14:PRINT@480,"":SOUNDI*15,1:NEXT:PRINT@228
,"THERE'S MORE COMING UP":FORI=1TO1500:NEXT
720 CLS0:POKE32104,LB:POKE32103,217+5-LB:RETURN
790 '*** HIGH SCORE ***
800 CLS0:HS=SC:PRINT@192,"YOU HAVE ACHIEVED A HIGH SCORE":PLAY"T10CDEFP5FEDCP5CD
EFP5FEDC"
810 PRINT@257,"ENTER YOUR INITIALS (UP TO 3)":PRINT@299,"";:INPUTN$:PRINT@303,BL
$+BL$;:IFLEN(N$)>3THEN810
820 RETURN
990 '*** BAT MOVEMENT ***
1000 FORI=31000TO31088
1010 READP$:POKEI,VAL("&H"+P$):NEXT
1020 DATA 8E,1,58,A6,84,81,DF,26,17,FC,7D,64,10,B3,7D,66,2F,6,FC,7D,66,FD,7D,64,
C3,0,1,FD,7D,64,20,21,8E,1,57,A6,84,81,DF,26,17,FC,7D,64,10,83,5,C0,2C,6
1030 DATA CC,5,C0,FD,7D,64,83,0,1,FD,7D,64,20,1,39,BE,7D,64,F6,7D,68,86,80,A7,80
,86,83,A7,80,5A,C1,0,26,F9,86,80,A7,80,39
1035 '*** CHECK IF SCREEN CLEARED ***
1040 FORI=32050TO32077:READP$:POKEI,VAL("&H"+P$):NEXT
1050 DATA 86,0,10,8E,7D,74,A7,A4,8E,4,20,A6,80,81,80,26,6,8C,4,C0,2F,F5,39,86,1,
A7,A4,39
1060 RETURN
1090 '*** SET UP WALL ***
2000 FORI=31200TO31236:READP$:POKEI,VAL("&H"+P$):NEXT
2010 DATA 8E,4,20,86,9F,A7,80,A7,80,A7,80,A7,80,A7,80,8C,4,FC,27,8,81,FF,27,EB,8
B,10,20,E9,A7,80,A7,80,A7,80,A7,80,39
2020 RETURN
```

# DRAGON CHESS

OASIS SOFTWARE



## *Some Dragons are bad losers!*




## *Others are just bad at losing.*

- Six levels of play.
- All legal chess moves including en-passant, castelling and pawn to queen promotion.
- List of previous moves stored which can be displayed or printed.
- Loading and saving of games positions from tape.
- High resolution graphics which can be flipped round to make black or white play from either end.
- Simultaneous text and graphics.
- Sides can be exchanged at any stage.
- Best move hint.
- Set up from any position.
- Opening move library.
- Moves may be taken back and play resumed from any point.
- Change level of play at any point in the game.
- Will adjudicate games between humans.
- Very high standard of play.
- Professional packaging and life- -time guarantee.

***DRAGON CHESS is available for just £9·95 from:-***

***OASIS SOFTWARE, Lower North Street, Cheddar, Somerset.***

ACCESS ORDERS TAKEN BY PHONE 24 HOURS A DAY

**0934 515265**

DEALER ENQUIRIES WELCOME

# Computer Connect 4 —matching wits with your Dragon

***R Chybalski** shows you how to build a game of strategy and challenges you to beat the Dragon at Connect 4.*

```
10 POKE65495,0
20 PMODE1,1:PCLS1:DIMG0(7,6):PCLEAR2
30 W1=7:W2=3:W3=15:W4=25:W5=12:W6=35
40 CLS:PRINT@10,"CONNECT 4"
50 PRINT:PRINT:PRINT"THE OBJECT OF THIS
GAME IS TO","GET FOUR OF YOUR OWN COUNTE
RS INA STRAIGHT LINE IN ANY DIRECTIONBEF
ORE YOUR OPPONENT. YOU TAKE  ALTERNATE T
URNS; COLUMNS BEING  FILLED STARTING AT
THE BOTTOM."
60 PRINT:PRINT"ONE OR TWO PLAYERS (1-2)?
 ";
70 A$=INKEY$:IFA$=""THEN70
80 IFA$="1"THENA5=1:PRINTA$:GOTO100
90 IFA$="2"THENA5=2:PRINTA$ELSE70
100 A7$=",40S8NU10NE4NH4":PZ=7:GZ=7
110 IFA5=1THEN120ELSEPRINT:PRINT"YELLOW
GOES FIRST":GOTO2080
120 X=RND(100):IFX>50THENPRINT"YOUR TURN
 FIRST!":K=2:GOTO1590
130 K=1:PRINT"COMPUTER'S TURN FIRST!":GO
TO1590
140 DRAW"BR3C2S8U6G3D1R4":RETURN
150 DRAW"BU1C2S8U4E1R2F1BD4G1L2":RETURN
160 DRAW"C2S8U6R4BD6L4U3R3":RETURN
170 DRAW"C2S8U6F4D2U6":RETURN
180 DRAW"C2S8BU1U4E1R2F1D4G1L2":RETURN
190 DRAW"C2S8BR2U6L2R4":RETURN
200 CIRCLE(236,26),8,4:PAINT(236,26),4,4
210 DRAW"C3BM"+STR$((22*PZ+38))+A7$
220 FORI=1TO7:T(I)=0:NEXTI
230 FORI=1TO7:F9=6
240 IFF9=0THEN270
250 IFG0(I,F9)=0THENF9=F9-1:GOTO240
260 IFG0(I,F9)=1THENT(I)=T(I)+1
270 SH=1:SV=1:SU=1:SD=1
280 F8=F9+1:IFF8>6THENT(I)=-1000:GOTO730
290 A=I
300 A=A-1:IFA<1THEN340
310 IFG0(A,F8)=1THEN340ELSESH=SH+1
320 IFG0(A,F8)=-1THENT(I)=T(I)+W1
330 GOTO300
340 A=I
350 A=A+1:IFA>7THEN390
360 IFG0(A,F8)=1THEN390ELSESH=SH+1
370 IFG0(A,F8)=-1THENT(I)=T(I)+W1
380 GOTO350
390 IFSH>3THENT(I)=T(I)+W2
400 A=F8
410 A=A+1:IFA>6THEN450
420 IFG0(I,A)=1THEN450ELSESV=SV+1
430 IFG0(I,A)=-1THENT(I)=T(I)+W1
440 GOTO410
450 A=F8
460 A=A-1:IFA<1THEN500
470 IFG0(I,A)=1THEN500ELSESV=SV+1
480 IFG0(I,A)=-1THENT(I)=T(I)+W1
490 GOTO460
500 IFSV>3THENT(I)=T(I)+W2
510 A=F8:B=I
520 A=A+1:B=B+1:IF(A>6ORB>7)THEN560
530 IFG0(B,A)=1THEN560ELSESU=SU+1
540 IFG0(B,A)=-1THENT(I)=T(I)+W1
550 GOTO520
560 A=F8:B=I
570 A=A-1:B=B-1:IF(A<1ORB<1)THEN610
580 IFG0(B,A)=1THEN610ELSESU=SU+1
590 IFG0(B,A)=-1THENT(I)=T(I)+W1
600 GOTO570
610 IFSU>3THENT(I)=T(I)+W2
620 A=F8:B=I
630 A=A+1:B=B-1:IF(A>6ORB<1)THEN670
640 IFG0(B,A)=1THEN670ELSESD=SD+1
650 IFG0(B,A)=-1THENT(I)=T(I)+W1
660 GOTO630
670 A=F8:B=I
680 A=A-1:B=B+1:IF(A<1ORB>7)THEN720
690 IFG0(B,A)=1THEN720ELSESD=SD+1
700 IFG0(B,A)=-1THENT(I)=T(I)+W1
710 GOTO680
720 IFSD>3THENT(I)=T(I)+W2
730 NEXTI
740 FORI=1TO7:F9=6
750 IFF9=0THEN770
760 IFG0(I,F9)=0THENF9=F9-1:GOTO750
770 F8=F9+2:IFF8>6THEN1020
780 A=I:SH=1:SV=1:SD=1:SU=1
790 A=A-1:IFA<1THEN810
800 IFG0(A,F8)=1THENSH=SH+1:GOTO790
810 A=I
820 A=A+1:IFA>7THEN840
830 IFG0(A,F8)=1THENSH=SH+1:GOTO820
840 IFSH>2THENT(I)=T(I)-W3
850 IFSH>3THENT(I)=T(I)-W4
860 A=F8:B=I
```

A GAME OF strategy is one that requires the players to think before making a move. The opponents' response to that move will be based on the move taken. Some strategy games are ones of pure skill, such as chess and draughts, others have an element of chance, such as backgammon.

Firstly you have to select a game and get to know the rules thoroughly. Then you must define the objectives for each move. In the game used in this article, Connect 4, these are to form a longer line and to frustrate your opponent's attempts to form longer lines. Obviously some sort of balance has to be struck between these. Most games can be analysed by calculating what is gained and lost by each possible move that a player can make (the overall benefit). You must also define when a player has won.

## Objectives

It helps to divide the objectives into individual blocks, as has been done in the Connect 4 program. It also helps if you program any benefit factors as variables which have their values assigned to them at the beginning of the program. This means that any changes in their value can be effected quickly.

The next stage is to draw a basic flow chart of the progress of play in the game. This should show each stage that the computer goes through when choosing its own move and receiving other players' moves. It should also contain a win testing routine.

The program can now be written, debugged and tested. Benefit factors generally require altering at the testing stage: you might find the computer making silly moves because it is giving too much prominence to some objectives — and silly moves make the computer easy to beat.

## Rules

There are two players in Connect 4 who take alternate turns choosing in which of seven columns to place a counter. Counters placed in a column form a stack which has a maximum height of six. The object is for a player to get four of his own counters in the frame to form a straight line in any direction (horizontal, vertical or diagonal) before his opponent.

The computer program accompanying this article allows two versions of the game: one for two players competing against each other, the other for one player to challenge the computer. It requires some skill to beat the computer (the computer's strategy can be altered as described below).

## Searches

In both versions the computer makes three different searches to assess which would be its best move. Essentially it has to try and form a line of four while defending against its opponent who is also trying to form a line of four. Three questions must be considered:

(1) Which move can I make to give the best chance for me to form a line of four?

(2) Which move must I make to prevent my opponent from forming a line of four immediately afterwards?

(3) If I make the move, will it let my opponent form a line of four when he places a counter on top of it?

A search method that can assess the game for all possible moves to the final outcome would be possible to write but it would require a large memory to store all the possible permutations of moves. It also might well produce a winning strategy which no player could beat if he went second. This would make the game very boring to play.

## Strategy

A better strategy from the playing viewpoint is to program the following for each possible move:

(1) Assess how many counters the computer can place in a row from that position, taking into account all previous counters that have been played.

(2) Assess how many counters the opponent can place in a row if he plays each move without the computer having made its next move. If the answer is four or more then the computer must play this move at all costs. If the answer is three it is a good idea to play it.

(3) Assess how many counters the opponent could place in a row if he plays on top of computer's next move. If the answer is four or more, this move must not be made. If the answer is three, it is an idea not ▶

◀ to make that move.

These have been written into the accompanying program, but the problem is how to balance the three to give the best result. To evaluate this an array is set up T(7) which contains a benefit score for each column. While going through searches, the computer adds to the benefit for a column in which it finds something to its advantage and subtracts if it finds a disadvantage. When it finishes searching, it takes the column with the greatest benefit score as being the best.

## Attacking moves

Lines 230 to 730 contain the program that enables the computer to find its best attacking move. This part is repeatedly run, once for each column. Lines 230 — 280 find out what depth a counter would reach if placed in that column. If the column is discovered to be full, the benefit score is made very large and negative for that column.

Lines 290 — 390 form a search for the number of potential counters that could be placed in a continuous horizontal line by the computer at that depth. For a potential line length, greater than three, it adds three to the score. For every counter already placed in that line it adds seven to the benefit.

Lines 400 — 500 form a similar search in the vertical direction, lines 510 — 610 in an upwards diagonal to the right direction, and lines 620 — 720 in an upwards diagonal to the left direction.

Lines 740 — 1020 contain the program which tells the computer whether making a move will allow its opponent to form a complete line of four counters. This part is again re-run for all seven columns. If making a move enables an opponent to form a continuous line of three by placing a counter above the computer's next move, 15 is subtracted from that column's benefit; if it enables a line of four, a further 25 is subtracted. Lines 740 — 780 find the depth of the counter above the one the computer would place. Lines 790 — 850 form a horizontal search, lines 860 — 930 form an upwards diagonal to the right, and lines 940 — 1010 form an upwards diagonal to the left search.

Lines 1030 — 1310 are the part of the program that enables the computer to ensure there are no obvious moves that it must make to prevent its opponent succeeding in the next play. If the opponent can make a line of three next move, 12 is added to the benefit for the column; a line of four gets an additional 35 added. Lines 1080 — 1140 perform the horizontal search, lines 1150 — 1220 perform the upwards diagonal to right, and lines 1230 — 1310 the upwards diagonal to left search.

## Best move

When the computer has completed the three searches the column with the maximum benefit value is taken as the best move (lines 1320 — 1410) and a win checking routine is entered. Upon return from the win checking routine, providing that the computer hasn't won, the player can enter his move by selecting the column required by moving the indication arrow and pressing ENTER (lines 1420 —1580). The program then proceeds to check if the player has won and returns to the computer's selection again.

## Checking

The win checking routine decides whether a player has won and, if one has, it finishes the game by declaring the winner. A search routine is used to determine whether any lines of four exist. Lines 1750 — 1800 are the horizontal search, lines 1810 — 1860 are the vertical search, lines 1870 — 1920 are the upwards diagonal to right, and lines 1930 — 1980 are the upwards diagonal to left searches. If no win is found, the computer exits using lines 1990 — 2010; if a win is found, lines 2020 — 2070 print out the winner.

## Changes

The rest of the program is relatively straightforward. Note that lines 10 and 2450 speed up and slow down the processor to get the computer thinking faster. The variables W1 — W6 in line 30 are the benefit quantities that are added or subtracted when the various situations described above are encountered.

If you wish the computer to attach more importance to obtaining its own line of four than to stopping its opponent from forming one, W1 and W2 should be increased. If you wish it to attach more importance to stopping its opponent, W3 — W6 should be increased. Changing any of these variables make the computer respond differently to various situations and may make it harder or easier to beat. ■

```
870 A=A+1:B=B+1:IF(A>6ORB>7)THEN890
880 IFG0(B,A)=1THENSU=SU+1:GOTO870
890 A=F8:B=I
900 A=A-1:B=B-1:IF(A<1ORB<1)THEN920
910 IFG0(B,A)=1THENSU=SU+1:GOTO900
920 IFSU>2THENT(I)=T(I)-W3
930 IFSU>3THENT(I)=T(I)-W4
940 A=F8:B=I
950 A=A+1:B=B-1:IF(A>6ORB<1)THEN970
960 IFG0(B,A)=1THENSD=SD+1:GOTO950
970 A=F8:B=I
980 A=A-1:B=B+1:IF(A<1ORB>7)THEN1000
990 IFG0(B,A)=1THENSD=SD+1:GOTO980
1000 IFSD>2THENT(I)=T(I)-W3
1010 IFSD>3THENT(I)=T(I)-W4
1020 NEXTI
1030 FORI=1TO7:F9=6
1040 IFF9=0THEN1060
1050 IFG0(I,F9)=0THENF9=F9-1:GOTO1040
1060 F8=F9+1:IFF8>6THEN1310
1070 A=I:SH=1:SV=1:SU=1:SD=1
1080 A=A-1:IFA<1THEN1100
1090 IFG0(A,F8)=1THENSH=SH+1:GOTO1080
1100 A=I
1110 A=A+1:IFA>7THEN1130
1120 IFG0(A,F8)=1THENSH=SH+1:GOTO1110
1130 IFSH>2THENT(I)=T(I)+W5
1140 IFSH>3THENT(I)=T(I)+W6
1150 A=F8:B=I
1160 A=A+1:B=B+1:IF(A>6ORB>7)THEN1180
1170 IFG0(B,A)=1THENSU=SU+1:GOTO1160
1180 A=F8:B=I
1190 A=A-1:B=B-1:IF(A<1ORB<1)THEN1210
1200 IFG0(B,A)=1THENSU=SU+1:GOTO1190
1210 IFSU>2THENT(I)=T(I)+W5
1220 IFSU>3THENT(I)=T(I)+W6
1230 A=F8:B=I
1240 A=A+1:B=B-1:IF(A>6ORB<1)THEN1260
1250 IFG0(B,A)=1THENSD=SD+1:GOTO1240
1260 A=F8:B=I
1270 A=A-1:B=B+1:IF(A<1ORB>7)THEN1290
1280 IFG0(B,A)=1THENSD=SD+1:GOTO1270
1290 IFSD>2THENT(I)=T(I)+W5
1300 IFSD>3THENT(I)=T(I)+W6
1310 NEXTI
1320 GZ=1:NN=1:F9=6
1330 IFNN>6THEN1360
1340 IFT(NN+1)>T(GZ)THENGZ=NN+1
1350 NN=NN+1:GOTO1330
1360 IFF9=0THEN1380
1370 IFG0(GZ,F9)=0THENF9=F9-1:GOTO1360
1380 F8=F9+1:G0(GZ,F8)=-1
1390 PAINT(38+22*GZ,194-22*F8),4,3
1400 DRAW"C2BM"+STR$((22*GZ+38))+A7$
```

```
1410 J9=-1:GOTO1750
1420 DRAW"C3BM"+STR$((22*GZ+38))+A7$
1430 CIRCLE(236,26),8,2:PAINT(236,26),2,
2:PZ=7
1440 DRAW"C2BM192"+A7$
1450 A$=INKEY$:IFA$=""THEN1450
1460 DRAW"C3BM"+STR$((22*PZ+38))+A7$
1470 IFA$=CHR$(13)THEN1520
1480 IF(A$=CHR$(8)ANDPZ>1)THENPZ=PZ-1:GO
TO1500
1490 IF(A$=CHR$(9)ANDPZ<7)THENPZ=PZ+1
1500 DRAW"C2BM"+STR$((22*PZ+38))+A7$
1510 GOTO1450
1520 IFG0(PZ,6)<>0THEN1500
1530 F9=6
1540 IFF9=0THEN1560
1550 IFG0(PZ,F9)=0THENF9=F9-1:GOTO1540
1560 F8=F9+1:G0(PZ,F8)=1
1570 PAINT(38+22*PZ,194-22*F8),2,3
1580 J9=1:GOTO1750
1590 PMODE1,1:PCLS3:COLOR1,3
1600 LINE(46,48)-(206,186),PSET,B
1610 PAINT(50,50),1,1
1620 FORX=60TO192STEP22
1630 FORY=62TO172STEP22
1640 CIRCLE(X,Y),8,3:NEXTY,X
1650 DRAW"BM74,14":GOSUB150
1660 DRAW"BM86,14":GOSUB180
1670 DRAW"BM98,14":GOSUB170
1680 DRAW"BM110,14":GOSUB170
1690 DRAW"BM122,14":GOSUB160
1700 DRAW"BM134,14":GOSUB150
1710 DRAW"BM146,14":GOSUB190
1720 DRAW"BM164,14":GOSUB140
1730 SCREEN1,0:IFA5=2THEN2090
1740 ONK GOTO200,1420
1750 D=1
1760 C=1:WI=0
1770 IF G0(C,D)=J9 THEN WI=WI+1 ELSE WI=
0
1780 IFWI=4THEN2020ELSEC=C+1
1790 IFC>7THEND=D+1:C=1:WI=0
1800 IFD>6THEN1810ELSE1770
1810 C=1
1820 D=1:WI=0
1830 IF G0(C,D)=J9 THEN WI=WI+1 ELSE WI=
0
1840 IFWI=4THEN2020ELSED=D+1
1850 IFD>6THENC=C+1:D=1:WI=0
1860 IFC>7THEN1870ELSE1830
1870 WI=0:C=1:D=4:C1=C:D1=D:GOTO1900
1880 C=C1:D=D1+1:D1=D:WI=0
1890 IFD>6THEND=6:C1=C1+1:C=C1:IFC>4THEN
1930
1900 IF G0(C,D)=J9 THEN WI=WI+1 ELSE WI=
0
1910 IFWI=4THEN2020
1920 C=C+1:D=D-1:IF(C>7ORD<1)THEN1880ELS
E1900
1930 WI=0:C=7:D=4:C1=C:D1=D:GOTO1960
1940 C=C1:D=D1+1:D1=D:WI=0
1950 IFD>6THEND=6:C1=C1-1:C=C1:IFC<4THEN
1990
1960 IF G0(C,D)=J9 THEN WI=WI+1 ELSE WI=
0
1970 IFWI=4THEN2020
1980 C=C-1:D=D-1:IF(C<1ORD<1)THEN1940ELS
E1960
1990 IFA5=2THEN2010
2000 IFJ9=-1THEN1420ELSE200
2010 IFJ9=-1THEN2250ELSE2090
2020 FORI=1TO1000:NEXT:PCLS
2030 SCREEN0,0:IFA5=2THEN2060
2040 IFJ9=-1THENPRINT"COMPUTER WINS!":GO
TO2410
2050 PRINT"YOU HAVE WON - I GIVE UP":GOT
O2410
2060 IFJ9=1THENPRINT"RED WINS!":GOTO2410
2070 PRINT"YELLOW WINS!":GOTO2410
2080 GOTO1590
2090 CIRCLE(236,26),8,2:PAINT(236,26),2,
2
2100 DRAW"C3BM"+STR$((22*PZ+38))+A7$:GZ=
7
2110 DRAW"C2BM192"+A7$
2120 A$=INKEY$:IFA$=""THEN2120
2130 DRAW"C3BM"+STR$((22*GZ+38))+A7$
2140 IFA$=CHR$(13)THEN2180
2150 IF(A$=CHR$(8)ANDGZ>1)THENGZ=GZ-1:GO
TO2170
2160 IF(A$=CHR$(9)ANDGZ<7)THENGZ=GZ+1
2170 DRAW"C2BM"+STR$((22*GZ+38))+A7$:GOT
O2120
2180 IFG0(GZ,6)<>0 THEN2170
2190 F9=6
2200 IFF9=0THEN2220
2210 IFG0(GZ,F9)=0THENF9=F9-1:GOTO2200
2220 F8=F9+1:G0(GZ,F8)=-1
2230 PAINT(38+22*GZ,194-22*F8),2,3
2240 J9=-1:GOTO1750
2250 CIRCLE(236,26),8,4:PAINT(236,28),4,
4
2260 DRAW"C3BM"+STR$((22*GZ+38))+A7$:PZ=
7
2270 DRAW"C2BM192"+A7$
2280 A$=INKEY$:IFA$=""THEN2280
2290 DRAW"C3BM"+STR$((22*PZ+38))+A7$
2300 IFA$=CHR$(13)THEN2340
2310 IF(A$=CHR$(8)ANDPZ>1)THENPZ=PZ-1:GO
TO2330
2320 IF(A$=CHR$(9)ANDPZ<7)THENPZ=PZ+1
2330 DRAW"C2BM"+STR$((22*PZ+38))+A7$:GOT
O2280
2340 IFG0(PZ,6)<>0THEN2330
2350 F9=6
2360 IFF9=0THEN2380
2370 IFG0(PZ,F9)=0THENF9=F9-1:GOTO2360
2380 F8=F9+1:G0(PZ,F8)=1
2390 PAINT(38+22*PZ,194-22*F8),4,3
2400 J9=1:GOTO1750
2410 PRINT"ANOTHER GO(Y-N)?"
2420 A$=INKEY$:IFA$=""THEN2420
2430 IFA$="Y"THENRUN
2440 IFA$="N"THEN2450ELSE2420
2450 POKE65494,0:END
```

# DESIGN AND WRITE GAMES FOR YOUR DRAGON!

## PROGRAMMING THE DRAGON FOR GAMES AND GRAPHICS

*Geoff Phillips*

This book explains the techniques of writing games for the Dragon computer. Using only BASIC, it contains a chapter on useful hints and tips for Dragon users — from editing and temporary line deletion to machine code programming.

★ The book describes an Adventure Game in detail, and builds up a complete working program.
★ There is an 'arcade' type of game — Star Battle — that is written in BASIC yet still moves fast enough to give a good game.
★ Details a typical 'real' application, in this case a stock control program dealing with the contents of a freezer.

ISBN 0 07 084703 7

**£5.95**

## DRAGON GAMES AND GRAPHICS

*Geoff Phillips*

This tape contains six programs for the Dragon 32. The programs are those developed in the above book. They are all written in BASIC, and can be examined and modified by the user if required.

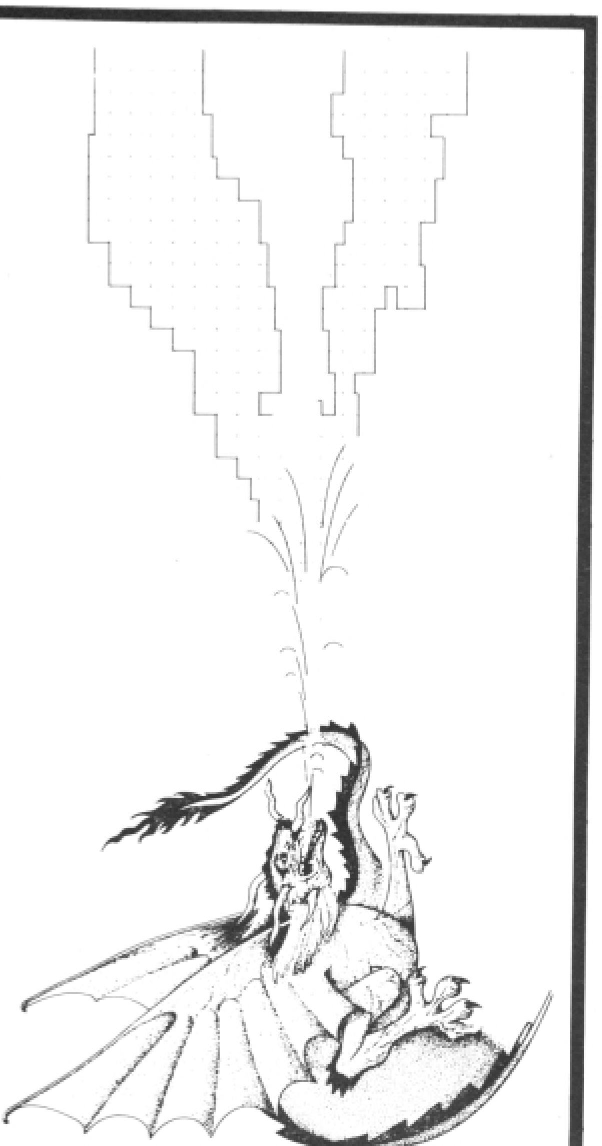
★ Six programs: BREAKOUT (with a difference)
CLOCK (with hands)
FREEZER (stock control)
ADVENTURE
CARD CHALLENGE (guess against the computer)
STAR BATTLE

★ All LISTable and in BASIC.

ISBN 0 07 084715 0

**£7.95**

This book and the accompanying software are available at all good booksellers, but in case of difficulty you may order them by just ticking the appropriate box(es). Then fill in your name and address and send this advert (or a copy of it) with your cheque or postal order to:



McGraw Hill

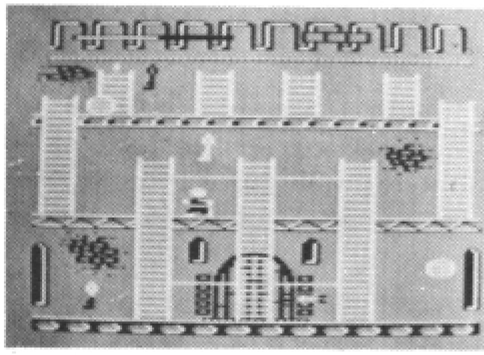
**McGraw-Hill Book Co. (UK) Ltd., FREEPOST, Maidenhead, Berkshire, SL6 2BU.**

Name: ..............................................................

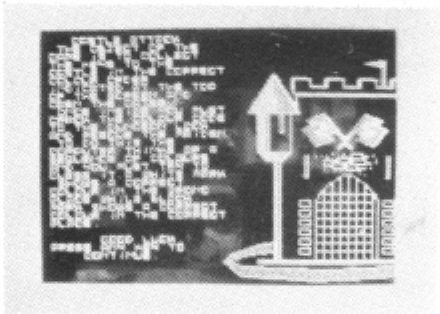
Address: ..............................................................

..............................................................

---

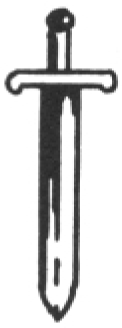


# M & D systems
# CASTLE ATTACK
## Dragon 32

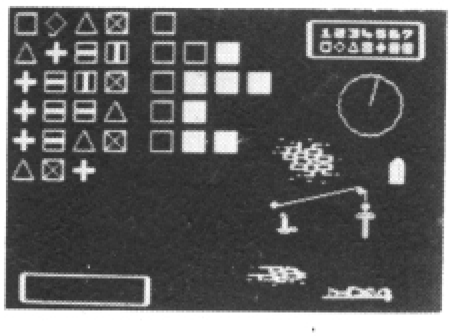


**All new superb MACHINE CODE game incorporating fantastic HI-RES COLOUR GRAPHICS AND SOUND**

Rush up and down the ladders testing every key to find the combination order to the castle lock. But TAKE CARE. If you loiter you may be destroyed by barrels of oil or even poisoned arrows. Once inside, crack the dungeon lock before the candle burns the rope that holds the sword above the maiden.



Features include: two different levels of play. Score and High Score presentation. Limited life line supply. Ever-increasing degree of difficulty. Full instructions. Extra graphics pages and three lives. No joysticks required.



*Dealer enquiries welcome*

# £6.50

Fully inclusive

Make cheque/POs payable to
**M & D Systems, 39 Sheridan Road Frimley, Surrey GU16 5DU**



# Hunt those bugs and kill them!

***Michael Whetstone** leads the search for the Dragon's bugs*

WHEN A BASIC program functions contrary to expectation, either by producing an error code or by resulting in the "wrong" answer, it should not generally prove too difficult to trace the bug and to edit the program accordingly. The Dragon is well equipped to handle such operations. If, however, only special input values cause the difficulty, and if careful checking of the program fails to reveal any obvious error, debugging can turn out to be an exasperating task.

As an example, Program A is a seemingly logical program, which nevertheless fails on rare (but predictable) occasions. The program is intended to factorise positive integers, or to determine that they are prime. The method used is to divide by 2 (line 60) and then by every odd number from 3 up to and, where applicable, including the square root of the highest odd factor (lines 70-100). The test applied for exact divisibility, as employed in lines 60 and 80, is to compare the quotient with its integer part, equality signifying a whole number.

As a modification to Program A lines 70 and 80 may be replaced by:

```
65 W = 3
70 FOR J = W TO SQR(N) STEP 2
80 IF N/J = INT(N/J) THEN W = J:
   PRINT J;: N = N/J: J = 2 ↑ 16: NEXT
   J: GOTO70
```

This modification greatly increases the program speed for higher numbers, except where these are perfect squares or primes, but as the principles relevant to the discussion are little affected, the simpler version of the program will be mainly discussed here.

This program will successfully factorise 651 as 3 7 31, or evaluate 52579 as being a prime number. If the method were to be followed using calculation by hand, it would prove effective for every positive integer above 1.

Despite this, there happen to be six numbers within the range 2-1000 which are factorised incorrectly. They are 289, 361, 578, 722, 841 and 961 (if the modified program is used, 867 must also be included in the list). Among the six, the odd numbers are the squares of 17, 19, 29 and 31 respectively, which the program wrongly concludes to be prime in each case. The even numbers are factorised incompletely as 2 289 and as 2 361, and so belong in the same category. Other squares provide no difficulty: 9, 25, 49, 121, 169 and 529 are all correctly factorised by the program. The rarity of this error, where less than 1 per cent of input values in the stated range are affected, could obviously result in such a program never being recognised as flawed.

## Test program

One means of investigating why certain values fail is to run the test program, Program B. This asks the computer directly whether it regards the result of applying its SQR function to the squares of the primes from 3 to 31 to be greater than, less than or equal to the true square root. Only 3 and 11 generate the anticipated read-out, eg, SQR(11*11) = 11.

Of the other eight primes, 5, 7, 13 and 23 give SQR(13*13) > 13, while 17, 19, 29 and 31 (the very values associated with failure of the program in Program A) produce SQR(17*17) < 17.

From this, the next move must be to find out what SQR(289) does equal, if not 17. The following results may be obtained (no program is required).

PRINT SQR(289) gives 17

PRINT 17−SQR(289) gives 7.4505806E−09

To anyone who is unfamiliar with the nature of the latter result, it may seem odd that an approximation which is ostensibly wrong in its first decimal place should be worked out to a further 7 decimal places, but this will become clearer shortly. For the moment, it is sufficient to see that the result of PRINT SQR(289), while given in a print-out as 17 (= 17.0000000), is nevertheless not identical to 17 and that the Dragon uses the result itself in making comparisons, and not the print-out version of that result.

Looking again at the operation of Program A, the position and mode of action of the bug become obvious. The FOR-...TO...STEP command in line 70 tests each possible divisor for N from 3 up to the square root of N, as determined by the SQR function. For N = 289, SQR(289) < 17 (as judged by the computer) and so the highest divisor tested is 15, not 17. As 289 is not divisible by any number in the range 2 to 15, it is given as prime.

Perhaps the simplest way in which to debug the program is to change line 70 as follows: 70 FOR J = 3 TO SQR(N)+1 STEP 2. Program A now tests 289 for division by 17 and successfully factorises it as 17 17. In the same way, all the other values which previously failed are correctly dealt with after this modification.

The only programming disadvantage that the change incurs is that for some numbers it will require one more test of divisibility than is absolutely necessary. For instance, 283 will be tested for divisibility by 17 using the modified form of line 70, whereas it is clear that the test is superfluous because 17 exceeds the square root of 283. A more complex modification can be made to reduce the likelihood of extra tests being performed. For instance, 70 FOR J = 3 TO SQR(N+(LOG(N)) STEP 2. This is largely a question of style. It is important, however, to avoid any correction which is effective over only part of the input range. An obvious example is: 70 FOR J=3 TO SQR(N+) STEP 2 ▶

```
10 REM FACTORIZE A POSITIVE              INTEGER, N
20 PRINT"NUMBER TO BE":INPUT" FACTORIZED";N
30 IF N<2 OR N>4294967295 THEN PRINT"OUT OF RANGE,REPEAT":GOTO20
40 IF N<>INT(N) THEN PRINT            "INTEGERS ONLY,REPEAT":GOTO20
50 IF N>3 THEN P=N ELSE P=1
60 IF N/2=INT(N/2) THEN N=N/2:        PRINT 2;:GOTO60
70 FOR J=3 TO SQR(N) STEP 2
80 IF N/J=INT(N/J) THEN N=N/J:        PRINT J;:GOTO80
90 IF N=1 THEN PRINT:GOTO110
100 NEXT J:PRINT N
110 IF N=P THEN PRINT"PRIME"
```

*Program A: seemingly logical but still occasionally fails*

```
10 CLS:PRINT" PROGRAMME TO CHECK RESULT        OF SQR FUNCTION"
20 DATA 3,5,7,11,13,17,19,23,29,           31
30 PRINT:FOR J=1 TO 10
40 READ Q:N=SQR(Q*Q)
50 PRINT" SQR(";Q;"*";Q;")   ";
60 IF Q<11 THEN PRINT"   ";
70 IF N<Q THEN PRINT"<";
80 IF N=Q THEN PRINT"=";
90 IF N>Q THEN PRINT">";
100 PRINTQ:NEXT J
```

*Program B: test program to see why some values fail*

◀ which factorises 289, 361, etc, and which does not test 283 for divisibility by 17, but which fails where large perfect squares are to be factorised — for example, 996475489, which is 31567 squared, the latter being prime. The run time of Program A for so high a perfect square is over five minutes, but the result can be predicted by the following method, analogous to Program B, but performed without a program:

LET N = 996475489
?SQR(N) – 31567 gives – 7.62939453E – 06
?SQR(N + 1) – 31567 gives – 7.62939453E – 06
?SQR(N) + 1 – 31567 gives .999992371
?SQRN(N + LOG(N)) – 31567 gives 3.2043457E – 04

Those functions giving negative answers when the true square root, 31567, is subtracted will inevitably fail for reasons explained above. It is clear that SQR(N+1) is as fallible as the original SQR(N) for high values of N.

## Practical method

The location and removal of the bug in Program A has followed an essentially practical method, suited to the specific program, and dependent upon the recognition that certain input values — quite rare values at that — gave anomalous results. This last point underlines the need for recognising the likely presence of the bug at the program writing stage, and this will prove easier if the underlying cause is identified. The minute difference between, for example, 17 and SQR(289) indicates that the difficulty lies in some sort of rounding or end error.

Program C seeks an answer to the question, "How does the Dragon respond when dealing with more than nine places of demical?" The fact that all print-outs show a maximum of nine places, for example, 123456789 or 66.6666667 or 9.87654321E–23, etc, might lead one to believe that 999999999 is the largest integer that can be operated on without first rounding the number. Program C shows that this is not the case, and that 4294967295 in fact is the number which holds that claim. This integer is the sum of all the powers of 2 from 0 to 31, and may be written as 2 ↑ 32 – 1. Above 4294967296, the minimum difference between two numbers that can be expressed

```
10 REM PROGRAMME TO TEST THE          DRAGON'S END-POINT ACCURACY
20 DATA 2147483,648,4294967,296,         8589934,592,17179869,184
30 FOR J=0 TO 3
40 CLS:READ X$
50 READ Y$:Y$=RIGHT$(Y$,3)
60 PRINT@4,"2 ^";31+J;" =";X$+Y$
70 PRINT@66,"X=":PRINT@78,"X-";    X$+Y$
80 FOR K=0 TO 10
90 Z=VAL(Y$)+K:Z$=RIGHT$(STR$(Z),3)
100 PRINT X$+Z$,VAL(X$+Z$)-VAL(X    $+Y$):NEXT K
110 IF J=3 THEN END
120 PRINT"PRESS C TO CONTINUE"
130 Q$=INKEY$:IF Q$="C" THEN         NEXT J
140 GOTO130
```

*Program C: investigates response to more than nine places of decimal*

is 0 or a multiple of 2. Program C shows that above 8589934591 (=2 ↑ 33–1), multiples of 2 ↑ 2 are involved, while above 17179869183 (=2 ↑ 34–1) differences are given as 0 or multiples of 2 ↑ 3. It is immediately seen that the limitation involved is directly based on binary: there can be no rounding of, say, a 10-digit decimal integer to 9 digits before its translation into binary code, or the integer limit would be a power of 10, not of 2.

This 32-column limit to exact integral operations is seen to exist whatever the magnitude of the numbers involved. From 2 ↑ 0 to 2 ↑ 32–1 the smallest difference measurable is 2 ↑ 0. From 2 ↑ 1 to 2 ↑ 33–1, the smallest difference becomes 2 ↑ 1, and so on. But what happens once the calculation crosses the decimal point?

This leads back to the earlier finding that ?17–SQR(289) = 7.4505806E–09. As the results found in Program C were powers of 2, could this also be a power of 2, this time a negative power?

?LOG(7.4505806E–09)/LOG(2) = –27

Therefore, 7.4505806E–09 is 2 ↑ –27. This test does not always work to such obvious effect because sums of more than a single power of 2 may be generated as an end error, in which case a non-integer results from the test LOG(N)/LOG(2). Program D, discussed below, analyses end errors while taking this possibility into account.

The Dragon, of course, uses the binary system in making calculations. The value stored within a variable remains in its binary form until converted into decimal notation by a PRINT statement or by the STR$ function. The binary value will be an approximation under circumstances where too many binary digits are required to give an exact answer: for instance, 1/7 in binary works out as 2 ↑ –3 + 2 ↑ –6 + 2 ↑ –9 . . . that is, an infinite series which the Dragon calculates as far as 2 ↑ –33. The other terms, starting at 2 ↑ –36, are inevitably ignored because of the impossibility of representing a recurrent binary fraction by a finite number of digits. The result ?17–SQR(289) = 2 ↑ –27 may be rationalised by noting that 17 can be exactly converted into binary, namely 10001, while the operation SQR(289) must therefore have resulted in the binary form 10000.111111111111111111111111111 (32 columns of binary) such that 10001 minus this figure gives 2 ↑ –27 as seen.

## Confirmation

A practical means in which to confirm these deductions is to take Program D, delete all the lines up to 90 (DEL–90), and to modify and extend the binary analysing loop as follows:

Change 100 R = 32: FOR J = 4 TO –128 STEP –1
Change 130 Q = Q – R: PRINT "2 ↑ ";J,

Then add a new line 90 which puts into memory Q the value or function value which is to be analysed. For example:

90 Q = SQR(289) or 90 Q = 17

Line 90 is changed by editing as required. Differences may be assessed directly; 90 Q = 17 – SQR(289). No PRINT or STR$ function intervenes to complicate the issue.

It has been seen that the binary equivalent of 17 and that of SQR(289) are each converted by the PRINT or STR$ command into the same decimal number, 17. It is no surprise that this is possible — 32 binary digits cover 2 ↑ 32 combinations ▶

```
10 CLS:PRINT@3,"PROGRAMME TO CALCULATE THE    INVERSE OF A NUMBER, N, AND TO
ANALYSE THE ROUNDING ERROR"
20 PRINT:INPUT"N";N
30 IF N<1 THEN CLS:PRINT"N<1,REPEAT":GOTO 20
40 PRINT"1/";N;"=";1/N
50 PRINT@224,"RESULT STORED IN COMPUTER MEMORYMINUS PRINT RESULT="
60 Q=1/N-VAL(STR$(1/N))
70 PRINT@303,Q:P$=LEFT$(STR$(Q),1):Q=ABS(Q)
80 IF Q=0 THEN END
90 PRINT"ANALYSIS OF ROUNDING ERROR IN    TERMS OF POWERS OF 1/2"
100 R=1:FOR J=1 TO 128
110 R=R/2
120 IF Q<R THEN NEXT J:END
130 Q=Q-R:PRINT P$;"(1/2)^";-J,
140 IF Q=0 THEN END ELSE NEXT J
```

*Program D: analyses end errors taking into account that they may just be the sums of more than a single power of 2*

◀ (including 0); nine digits of decimal cover just 1E9, such that an average about 4.3 binary numbers will be converted to the same decimal equivalent. The reverse process, as governed by INPUT or VAL, is less ambiguous: a single 32-digit binary number will be generated from any 9-digit decimal input. This is no guarantee that no approximation will be involved. The mathematical nature of base 2 to base 10 interconversion is such that only (+ or −) powers of 2 and sums of these remain free from approximations. On the Dragon, this exact conversion will cover all decimal integers up to 2↑32−1. Most decimal fractions must, on the other hand, be approximated as described for 1/7. That a very real difference exists between an integer such as 4, and a decimal fraction such as .4, is clearly shown by the following:

PRINT (1 + 2 + 4) − (1 + 4 + 2)    gives 0
PRINT (.1 + .2 + .4) − (.1 + .4 + .2) gives 2.32830644E − 10

Step-by-step analysis of the second sum may be performed by means of the binary analyser, derived from Program D in the previous paragraph.

It should now be clear that a difference may exist between a variable, N, and the value that would be obtained by first converting to decimal (by PRINT or STR$) and then back to binary (by INPUT or VAL). One reason that this difference is important, is that it may be disguised by there being a single decimal version for both values, such that the tracing of bugs is hindered by the lack of visible signs. Program D is designed to estimate such differences directly by means of the double function VAL(STR$()), and to analyse this in terms of one or more powers of 2↑−1, or 1/2. Simple modification of the program will supply data for functions other than /, which is the function used for demonstration purposes in the program in its present form. Used in conjunction with the binary analyser derived from it, Program D is a powerful tool for seeing through the eyes of the Dragon.

## Precautions

Where, then, should one take precautions to avoid the unwanted effects of end errors during programming? One place is in the FOR ... TO ... STEP command, even where no special functions such as SQR are involved. For example:

```
10 FOR J = 0 TO 1 STEP .1
20 PRINT J: NEXT
```

The value 1 is never printed. After the program has been run, the true binary value of J may be checked, allowance being made for .1 having been added on exiting the FOR. . .NEXT loop. ?J−1 gives 4.65661287E−10 (=2↑−31).

Correction is straightforward once the problem is seen; 10 FOR J = 0 TO 1.01 STEP .1

Integral values used in the STEP function cause no problems, except in the unlikely event that J should exceed 2↑32−1.

Usually only comparison statements are affected critically by end errors: IF. . .THEN. . .ELSE, ON. . .GOTO/GOSUB, and FOR. . .TO. . .STEP should only be used with special functions such as SQR where safeguards have been taken. The numerical functions INT,FIX and SGN can be badly affected, as can a number of commands which contain an implicit INT function, eg, SET, RESET and several high resolution commands such as PSET, CIRCLE etc. All may be similarly corrected.

Where correction is not so simple, the design of the program may be modified.

Program D needs greater accuracy than is provided by the power function, ↑, and so a FOR. . .NEXT loop with repeated division by 2 is used in its place. Likewise, there are numerous ways in which string functions can be used to provide greater accuracy, or to calculate results to a greater number of decimal places than is normally possible. To a very minor extent, Program C reflects this in its storage and combination of 10- or 11-digit decimal numbers using strings, so that the subsequent conversion to binary by VAL can be investigated.

Finally, it should be stressed that number crunching on the Dragon can be highly successful as long as one never forgets that the computer is dealing with a series of approximations dictated by the digital limitations of its processor. These approximations are so close that they make no more than the most marginal difference to most computations, but they can prove crucial when making comparisons or using INT−type functions. ■




# of the hill!

*The RAINBOW* is the biggest and best magazine available for the TRS-80® Color, TDP-100, MC-10 and Dragon-32 Computers.

And no wonder! It's over 300 pages thick each month . . . pages brimming with programs, product reviews, tutorials, columns, hints and tips about *your* computer. Yes, it is considered "the" Color Computer magazine to buy.

Don't delay. For only $22, you can get *the Rainbow* every month of the year. Then *your* CoCo will be Kong of the Hill too!

**U.K. Subscription Rates**
**U.S. $57 surface rate**
**U.S. $85 air rate**

© 1983 The Rainbow

® Registered trademark of Falsoft Inc ® Registered trademark the Tandy Corp

**the Rainbow** 9529 U.S. Highway 42
502/228-4492 P.O. Box 209
Prospect, Ky. 40059

YES! Sign me up for a year (12 issues) of **the RAINBOW**.

Name ______
Address ______
City ______ State ______ Zip ______
Payment Enclosed
Charge VISA MasterCard American Express
My Account# ______ Interbank# (MC only) ______
Signature ______ Card Expiration Date ______

Subscriptions to **the RAINBOW** are $22 a year in the United States. Canadian and Mexican rate U.S. $29, surface rate to other countries U.S. $57, air rate U.S. $85. All subscriptions begin with the current issue. Please allow up to 5-6 weeks for first copy.

Just for your
TRS-80® COLOR
TDP System 100
Dragon-32
MC—10

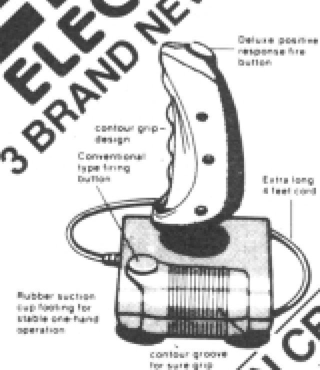
# ELKAN ELECTRONICS

## 3 BRAND NEW PRODUCTS

**Quick-Shot Joystick**

Specially developed by Elkan Electronics for the DRAGON 32 or Tandy Color Computer. Plugs straight into the machine with NO NEED for expensive adaptors or tamers. Improve your scores!

• ELKAN PRICE £14.95 (+ £1.00 p&p) EACH or £28.95 (+ £1.50 p&p) PAIR




**DRAGON CRUNCHER**

First cassette-based menu-driven program to convert most DRAGON 32 programs to Tandy color - and most tandy color programs to DRAGON 32!! Easy conversions for RAINBOW, COLOR COMPUTER NEWS, COLOR COMPUTER MAGAZINE, HOT CO CO.

1st

ELKAN PRICE £7.95 p & p free

NOW HERE

Quick-reference card for DRAGON 32. Easy to handle! Easier to use than the manuals! Why waste time and frustration struggling with the books? This card has it all! *Contains hitherto unpublished information on the DRAGON 32! You have to see it to believe it!

ELKAN PRICE £3.95 (p & p free)

NANO SYSTEMS CORP.

For all your DRAGON'S requirements, send cash, cheque, postal/money order or credit card number to:

**ELKAN ELECTRONICS, FREEPOST, 11 BURY NEW ROAD, PRESTWICH, MANCHESTER M25 6LZ or Telephone 061-798 7613**

Prestel number 258-2022. Export and trade enquiries welcome.

# GARLAND COMPUTING

35 DEAN HILL, PLYMOUTH PL9 9AF
TELEPHONE: 0752 41287

# EDUCATIONAL SOFTWARE FOR THE DRAGON

**LEARNING MATHS** A series of programs for ages 9-12. Each package contains 3-4 programs, many with animations and entertaining games, to help learn the principles of maths. Used in schools throughout the country (each package £7.00)

- Angles
- Directed Numbers
- Fractions
- Co-ordinates and Lines
- Symmetry
- Motion Geometry

**PHYSICS** A set of programs for use in schools and at home — animations and simulations to help learn about important topics in secondary school physics.

- Ohm's Law (£9.00)
- Principles of the DC Motor (£9.00)
- Circuits — switches (£11.00)
- Circuits — diodes (£11.00)

**PATTERNS** A unique program enabling you to create repeat patterns for use in decoration or for the design of knitting stitches (£5.95)

Prices include VAT and P&P. Available by mail order, or from selected computer shops and educational suppliers. Please send for full details.

## the educational specialists

# Place your Dragon in charge

***Dennis Riley** shows how to put a standard cassette recorder under the control of your Dragon 32*

THIS ARTICLE AND the subsequent changes in my cassette recorder, were prompted by the purchase of David Lawrence's excellent book, *The Working Dragon 32.*

This book revealed to me the magic and endless possibilities of the 'DATA FILES', up until now little used and, unfortunately, ill-defined in the manual.

However, I was still at a loss. For while awaiting delivery of my Dragon, I had bought a cassette recorder without the remote control facility. Why didn't the advertising leaflet tell me that remote control was necessary?

Naturally I had tried some data filing, but even with Pre-recorded material I was rewarded with the ominous IE/ERROR message.

What to do? I was fully aware that the need for filing systems would eventually become pressing and that I was missing out on what is one of the Dragon's most powerful features.

Having checked the price of the necessary cassette recorder and finding it out of my range, necessity became the mother of invention.

I found that all that was required was a little soldering, an inline jack plug and socket (these can be purchased for a little over 50p), a couple of lengths of wire and a little time.

## Not recommended

Before I go into details, I would point out that this conversion is not recommended for new rack systems.

I remembered from my days as an audio and TV engineer that the first cassette recorders with remote control had a switch on the microphone which turned the machine on or off during record or playback. In fact, all this switch did was to break the HT line to the motor, thus stopping the machine.

Surmising that the "clicks and clacks" from inside the computer were just the opening and closing of a switch, I proceeded to do the following:

1. Remove the back from the cassette recorder (usually there are four screws, one normally in the battery compartment). If you haven't been inside a cassette recorder before, be careful when separating the two halves of the machine as there are quite a few wires floating about.
2. Locate the motor, a round thing with a small pulley on it and two wires leading to it.
3. Trace the two wires, one of which should go to a point on the printed circuit board. This is the one we want, mainly because it is more convenient than the other (Figure 1).
4. Desolder this wire from the pcb, taking care not to burn any of the other wires. In its place solder one of the wires.
5. Solder the other length of wire to the lead you removed from the pcb. It is advisable to wrap a little insulating tape around this joint (Fig II).
6. The cassette recorder can now be put back together, although it may be necessary to file a little slot on the casing to allow the wires out. I put mine just above the mike and earphone sockets (Fig III).
7. All you need to do now is connect the two lengths of wire into the jack socket (I needed to change the plug-on cassette lead supplied with my Dragon) and then the jack plug can be inserted — and the Dragon is in control (Fig IV).

If you do this conversion, remember to disconnect the cassette recorder from the mains as the motor works off only 6 volts. The mains work off 250 volts.

Now that you have it taped, so to speak, try CLOADing or CSAVEing, a program. Connect your cassette recorder back up to the mains, pop in a cassette tape, press play or play/record, type in CLOAD"PROGRAM" or CSAVE"PROGRAM" then press ENTER. ▶

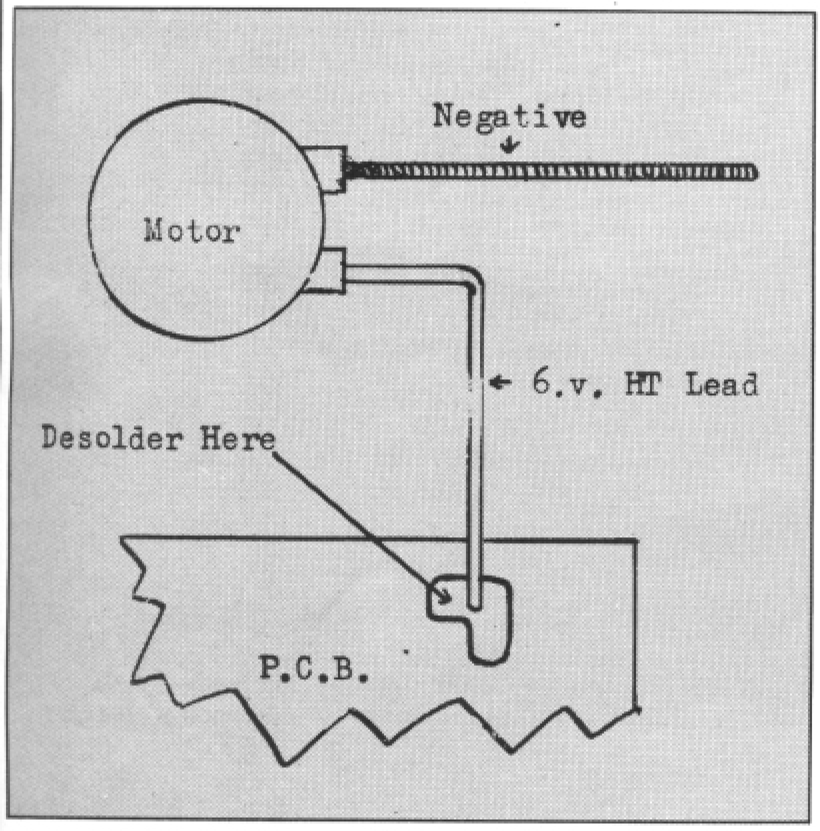


*Figure I: the wire to the printed circuit board is the most convenient*

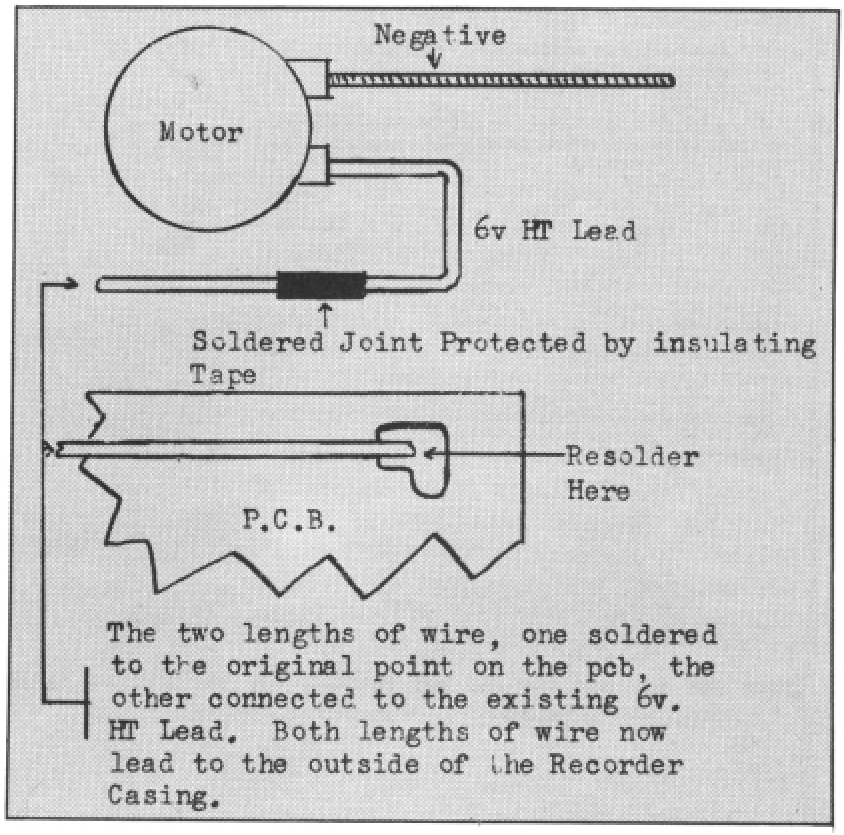


*Figure II: the two wires in position, both leading outside the recorder*

◀ Dragon now takes over and starts the cassette recorder and stops it when it has finished the operation. Great, isn't it? Oh dear! Can't you rewind? This is easily solved by entering as a direct command MOTORON. The cassette recorder is now returned to your control and you can rewind, erase, playback, whatever, while you are entering your programs. Switching off the computer automatically cancels the MOTORON command. The direct command MOTOROFF returns control of the cassette recorder once more to your Dragon.

## Useful command

Another useful direct command is the little used AUDIO ON. If this is entered when the computer is switched on, by increasing the volume on your TV set slightly you can actually listen to the cassette recorder and the computer chatting away merrily to one another during CLOAD and CSAVE. It is also useful during the CLOAD operation in as much as you can fast forward the cassette tape, play until you hit a blank spot and then CLOAD without getting an I/O ERROR message. This is because you started CLOADing in the middle of a program.

Now that you have your Dragon under control, and the Dragon has the cassette recorder under control, the wealth and abundance of data filing are yours. I fully recommend the previously mentioned book — not because I know the author; I don't — but because of its simple and straightforward introduction to data filing.

## Changing back

Should you at any time need to use your cassette recorder for something other than Dragon operations, then the simple expedient of a spare jackplug with both of the internal connections shorted together and inserted in the jack socket, will release your cassette recorder from the Dragon's grip.

The following short sample program shows another use for a Dragon in control. I use it on all my cassette tapes, and it saves the problem of remembering which tape has which programs on it, and also cuts down on CLOADing errors. I find it invaluable on the C120s that I use as master tapes and on which I duplicate all

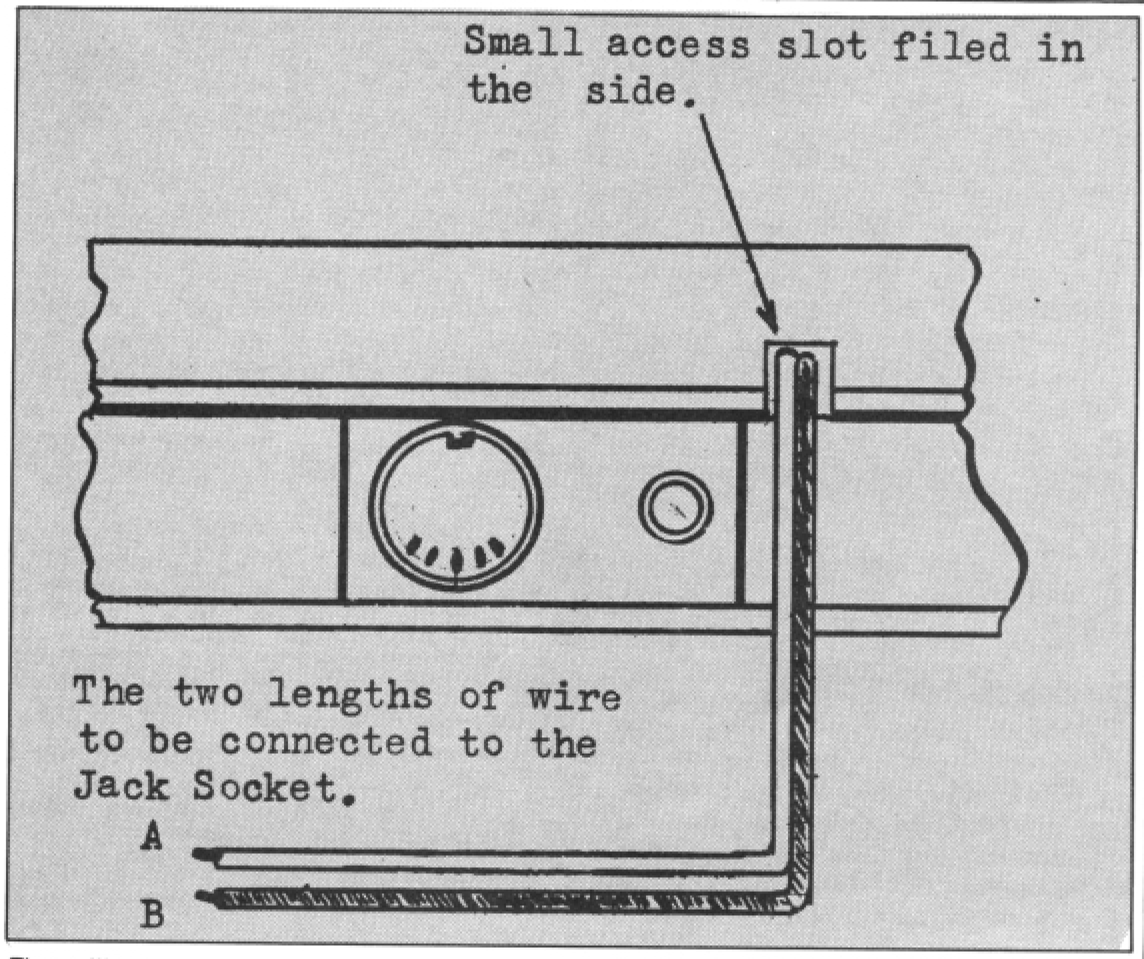



*Figure III: file an access slot for the wires when putting the recorder back together*

the programs I have sweated over in case of an accident to the C15s I use for normal operations. Only the first 20 or 30 seconds of the cassette tape need be saved for it.

```
5 CLS
10 PRINT@235, "DIRECTORY":FOR D = 1 TO
   1000:NEXT D:CLS
20 PRINT "PROGRAMS ON THIS TAPE"
30 PRINT
40 PRINT " 1) ALIEN ATTACK"
41 PRINT " 2) MILLIPEDE — DRAGON USER NO.
   II"
42 PRINT " 3) TOPSY — M/C ROUTINE"
43 PRINT " 4) DUCKSHOOT-DRAGON USER NO.
   II"
44 PRINT " 5) BRAINBOX-DRAGON USER NO. II"
45 PRINT " 6) DEATH WALL-PCW.VOLII,14,P8"
80 INPUT "ENTER NUMBER OF PROGRAM AND
   PRESS 'ENTER' ";Q: ON Q GOTO 100, 200, 300,
   400, 500, 600
100 CLS:PRINT "loading alien attack":SCREEN-
   0,1:CLOAD"ALIEN"
200 CLS:PRINT "loading millipede":SCREEN-
   0,1:CLOAD"MPEDE"
300 CLS:PRINT "loading topsy":SCREEN-
   0,1:CLOAD "TOPSY"
400 CLS:PRINT" loading duckshoot":SCREEN-
   0,1:CLOAD "DUCKS"
500 CLS:PRINT "loading brainbox":SCREEN-
   0,1:CLOAD "BBOX"
600 CLS:PRINT "loading death wall":SCREEN-
   0,1:CLOAD "DWALL"
```

This program can be adapted to your own use. In fact, the example titles were used to fill out the program to show what can be done.

## Program notes

Lines 5-30 Set up titles.

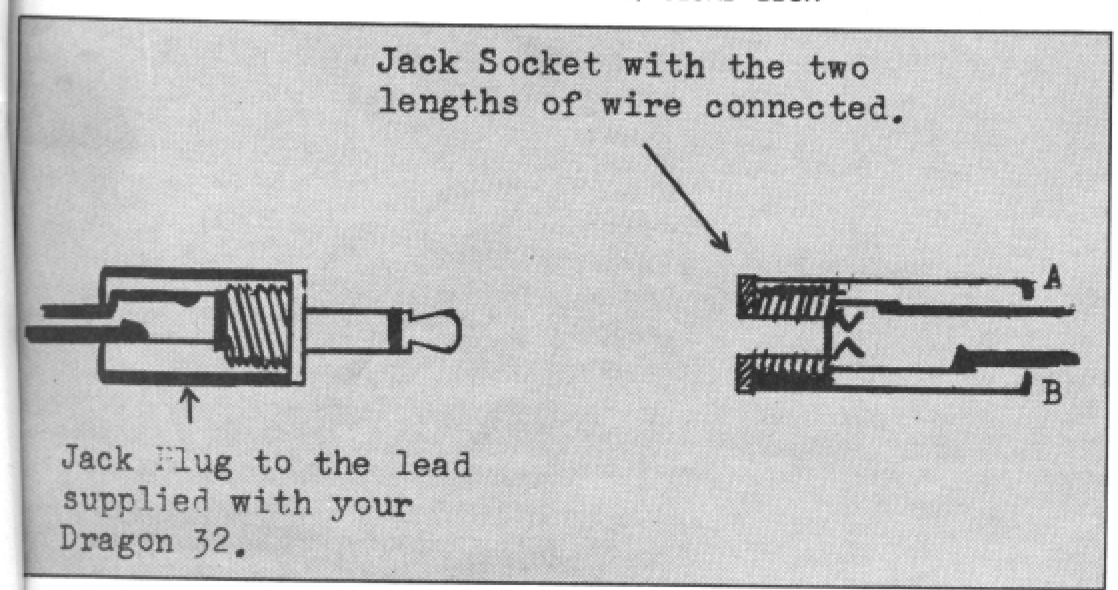
Lines 40-45 List the contents of the cassette tape. Note that the comments can be of any length and can contain as much information as you like. What is important is the number "1)" etc. I have only used six titles in this example, but should you use more than 10, the addition of another line — 70 INPUT" PRESS 'ENTER' TO CONTINUE";Q — will allow you to scroll the next batch of titles.

Line 80 A simple ON/GOTO statement sends the program to the operational line on the INPUT Q.

Lines 100-600 These lines display on the screen the title of the program being loaded. The screen is black on orange during search and load and once the program is loaded the screen reverts to normal black on green. CSAVE the proram "DIR". The CLOAD titles must be the ones they are CSAVED under.

Now simply by inserting a cassette tape into the recorder, pressing down the play button, enter CLOAD only (On this command the Dragon loads the first program on the tape). With the play button still down, RUN the program, enter the number required and let the Dragon do the rest of the work.

This program does save time as it cuts down on the chance of the Dragon being told to CLOAD a program that isn't on the tape (that is, CLOAD"DWELL" instead of CLOAD"DWALL"). Happy programming.■




*igure IV: join the wires to the jack socket, insert the jack plug, and the Dragon is in control*

# OPEN FILE FOR DRAGON USERS

**Send us your Dragon programs, beginning with a general description and then explaining how the program is constructed. Take care that the listings are all bug-free, enclosing a printout if possible. We pay £6 for each bug-free program published, double for the program of the month. If you have any problems with the listings, please send your queries to the appropriate author, *Dragon User,* 12-13 Little Newport Street, London WC2R 3LD.**

## INSTR

*From K Garwell in Stoke-on-Trent*

THE INSTR INSTRUCTION offers a neat alternative to a number of IF — THEN statements in many cases, particularly in replies to multi-choice questions. A graphics program that I am developing offers a number of menus and I will use one of these to illustrate my point.

A menu appears as follows:

ACTIONS ARE
1. MOVE
2. DRAW
3. ERASE
4. PAINT

Any one of the four actions could be selected by code such as line 1000 on where www,xxx,yyy,zzz represent the entry points to the move, draw, erase and paint routines.

```
1000 INPUT N:ON N GOTO www,xxx,yyy,zzz
1010 PRINT"INCORRECT SELECTION":GOTO1000
1020 REM MOVE,DRAW,ERASE,PAINT CODE
1030 REM
1040 REM
2000 IC$=INKEY$:IF IC$="" THEN 2000
2010 ON INSTR(1,"1234",IC$) GOTO www,xxx,yyy,zzz
2020 ON INSTR(1,"MDEP",IC$) GOTO www,xxx,yyy,zzz
2030 PRINT"INCORRECT SELECTION":GOTO2000
2040 REM MOVE,DRAW,ERASE,PAINT CODE
```

The disadvantage of this code is that it requires the number to be typed and then the enter key to be pressed and it is much more convenient to be able to make the selection by pressing only the number key. It is even more convenient, and much easier to remember if the first letter of the required action can be used. So that in the example given one can type either 1,2,3,4 or M,D,E,P. Now the INSTR instruction comes in to its own.

Line 2000 on shows the alternative code to line 1000 on, but with the features mentioned. Line 2000 waits for the key to be pressed. Line 2010 uses INSTR to generate 1,2,3 or 4 if a number key was used and goes to the appropriate code. If a number key was not used INSTR in 2010 will return 0 and the GOTO will drop through to the next line.

Here at line 2020 if one of the four letters were used, again 1,2,3 or 4 would be returned by INSTR resulting in going to the appropriate code. If an incorrect key was used this line would return 0 and drop through to line 2030 which would put up the INCORRECT SELECTION message and return to the input line.

## Speed POKE

*From Simon Locke in Ludlow*

AS A SEEMINGLY large number of Dragon 32 owners have had problems with programs which contain the speed-up POKE, I have produced this short routine. It is entered manually above the loaded program and when run using its own first line number, searches the loaded program for the relevant POKEs.

It prints these on the screen and printer, if available, in the form of LINE 10 65495 or LINE 10 &HFFD7. Once the program has been read completely, FINISHED appears on the screen. Then the POKEs can be deleted from the program, the search program deleted and the edited version of the loaded program saved on tape.

**Variables**

S — Start of the Basic program
E — End of Basic program
LP — Memory pointer for next line
LN — Present line number
P — If '1' then prints lines: if '0' displays on screen only
LE — Number of first line in search program (this saves time as the loaded program only is read and not the search program as well)

**Stages**

1 List loaded program for last line number.
2 Enter search program with line numbers above loaded program.
3 Run search program.
4 Make a list of line numbers with POKE if no printer available.
5 Delete lines of search program.
6 Edit POKEs from the lines listed.
7 Resave the loaded program.

```
10000 CLS          10010 PRINT"PRINTER AVAILABLE (Y/N)?"
10020 A$=INKEY$:IFA$=""THEN10020:ELSEIFA$="Y"THENP=1:ELSEIFA$="N"THENP=0:ELSE100
00
10030 PRINT"NUMBER OF FIRST LINE IN SEARCH?":INPUTLE
10040 CLS:S=PEEK(25)*256+PEEK(26):E=PEEK(27)*256+PEEK(28)-2
10050 LP=PEEK(S)*256+PEEK(S+1):LN=PEEK(S+2)*256+PEEK(S+3)
10060 FORA=S TO LP:IFPEEK(A)=147 THENGOSUB10100                         10070 NEXT
10080 S=LP:IF LN=LE THENPRINT"FINISHED":ELSEGOTO10050                    10090 END
10100 IFPEEK(A+1)=38 AND PEEK(A+2)=72 AND PEEK(A+3)=70 AND PEEK(A+4)=70 AND PEEK
(A+5)=68 THENGOTO10150
10110 IFPEEK(A+1)=54 AND PEEK(A+2)=53 AND PEEK(A+3)=52 AND PEEK(A+4)=57 THENGOTO
10120:ELSERETURN
```

Continued on page 43

```
10120 IFPEEK(A+5)=53 THENPRINT"LINE ";LN;" 65495":IFP=1 THENPRINT#-2,"LINE ";LN;
" 65495":RETURN:ELSERETURN
10130 IFPEEK(A+5)=52 THENPRINT"LINE ";LN;" 65494":IFP=1 THENPRINT#-2,"LINE ";LN;
" 65494":RETURN:ELSERETURN
10140 RETURN
10150 IFPEEK(A+6)=55 THENPRINT"LINE ";LN;" &HFFD7":IFP=1 THENPRINT#-2,"LINE ";LN
;" &HFFD7":RETURN:ELSERETURN
10160 IFPEEK(A+6)=54 THENPRINT"LINE ";LN;" &HFFD6":IFP=1 THENPRINT#-2,"LINE ";LN
;" &HFFD6":RETURN:ELSERETURN
10170 RETURN
```

# String manipulation

*From Paul Yeandle in Poole*

DID YOU know that the string function MID$ can be used in two opposite ways? To illustrate, I will use A$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ".

Dragon users will probably be familiar with X$ = MID$ (A$,5,3) which extracts three characters from A$ starting at the fifth. So X$ = "EFG". They will almost certainly not be aware of the opposite use because there is no reference to it in the reference manual. MID$ (A$,5,3) = X$ will replace three characters in A$ with the first three of X$. So, if X$ = "12345" then A$ will become "ABCD123HIJKLMNOPQRSTUVWXYZ". This useful facility means that strings can be amended without the complex breaking-up and re-joining process which is necessary on some other machines.

The routine below prints four hands of cards chosen randomly from a standard 52-card pack. It uses both forms of the MID$. Firstly, two strings of 52 characters are set up to represent the cards: C$ contains the face values "234567890JQKA" repeated four times (note the single character "0" to represent the 10), S$ contains the four suits (lines 100-120). The names of the four hands are set up at line 130.

For each card in each hand, a random number in the range 1-52 is selected and the character in that position of the face value string C$ is extracted into X$ (line 180). Once a card has been chosen, its face value is "spaced out" of string C$ (line 220) so it is necessary to check whether a card has already been dealt (line 190). Since 10 is represented by a "0", it is necessary to convert this into "10" (line 200). A more sophisticated routine can be written which does not need to test for cards already dealt.

Note that LEFT$ and RIGHT$ do not have this inverse facility so that LEFT$ (A$,5) = X$ will result in a syntax error.

```
100 C$="234567890JQKA"
110 C$=C$+C$+C$+C$
120 S$=STRING$(13,"C")+STRING$(13,"D")+STRING$(13,"H")+STRING$(13,"S")
130 H$="    NORTHEAST SOUTHWEST"
140 FOR H =1 TO 4
150 : PRINT MID$(H$,H*5,5)
160 : FOR C= 1 TO 13
170 :    R=RND(52)
180 :    X$=MID$(C$,R,1)
190 :    IF X$=" " THEN 170
200 :    IF X$="0" THEN X$="10"
210 :    PRINT X$;MID$(S$,R,1);"  ";
220 :    MID$(C$,R,1)=" "
230 : NEXT C
240 : PRINT:PRINT:PRINT
250 NEXT H
```

# Munchy

*From Alison Dowey in Northern Ireland*

IF YOU LIST out many Dragon games programs, you will notice that two Basic statements are ever present, PEEK and POKE. This game provides you with a simple but thorough explanation of what PEEK and POKE do, and how you can make use of them.

The first thing to understand about PEEK and POKE is what they do. To POKE something is simply to slot it into a certain position in the memory. Using PEEK does exactly the opposite, it looks at a memory position and tells us what it finds. The form of PEEK and POKE are as follows: POKE X,Y — where X is the memory position you are altering, and Y is the value you are placing in it; Y=PEEK(X) — where X is the memory location being examined and Y will then be the value of this location's contents. This all sounds a bit complicated, but don't be put off.

The first thing you need to know is that the memory locations of the text screen (ie the screen that your text appears on) are between 1024 and 1535 and the spaces are progressively numbered identical to the print screen. Therefore, to POKE 1025, 159 is the same as PRINT @ 1,CHR$(159), and its great advantage over PRINT @ is the speed of execution. To get around working the number out every time, do the following:

```
10        B=1024; POKE B+224,159
```

When run you get a yellow block at position 224 on the screen.

Now that you know the basic POKE, here's a simple movement. Type in the following:

```
10        B=1024
20        A=B+239
30        POKE A,159
40        A$=INKEY$: IF A$="" THEN
          40
50        IF A$="Z" THEN A1=-1
60        IF A$="X" THEN A1=1
70        IF A$="M" THEN A1=32
80        IF A$="K" THEN A1=-32
90        A=A+A1
100       GOTO 30
```

When run, you will see a yellow square. Press the keys Z,X,M and K. You can now mechanically move the square.

Now add:

```
15        CLS0
95        IF A<1024 OR A>1024+511
          THEN A=A-A1
```

Line 95 stops you from POKEing into other sections of the memory as to do this can lose your program.

As you can see the movement is ▶



slow and mechanical and we'll alter this now. Change line 40 to A$=INKEY$. Now run it and see the difference. There is one problem left, it has a trail. To get rid of this we need to black out the last position of the square. Type in:

```
99        POKE A,128
```

You now have a complete movement routine which will prove very useful in your own programs.

Two more lines show what PEEK can do:

```
25        POKE B+RND(511),207
98        IF PEEK(A)=207 THEN SOUND 55,1: POKE B+RND(511),207
```

This adds the eating principle to your newly acquired repertoire.

**Program notes**

| Lines | |
|---|---|
| 360 | Sets the data in a position to be read |
| 370 | Sets all the variables: B=start of screen in memory, M1, M2 and M3 are the starting positions of the three ghosts, M5, M6 and M7 are the directions the ghosts start in, PA=your position at start, DO=number of dots on screen at start |
| 390 | Reads the 479 data figures which draw the screen |
| 400 | If the figure is a 1 then a blue wall is drawn (175) else a yellow dot (145) |
| 410 | Draws the four energisers (223) where a 2 is in the data |
| 420 | Sets up a loop |
| 430-470 | Hold data |
| 480 | Tidies up screen at start by erasing a few dots |
| 490 | Gives a short interlude |
| 500 | Sets up main play loop using TY. Blacks out the last position of your pac-man by using (128), a black square. The remainder of the line checks to see if you are about to eat a yellow 145 and if so it adds one to the score, prints the score and takes one off DO, the number of dots remaining on the screen. |
| 510 | Checks for a screen clearance |
| 520 | Draws your new position |
| 530-560 | Takes the next direction from the keyboard |
| 570 | Checks whether or not you can go in that direction and if so transfers your chosen direction PW into PO, the direction your man moves in. If you can't go that way at the minute, you will as soon as you meet an opening in that particular direction, a very useful trick. |
| 580 | Stops you if you come to a wall |
| 590 | Checks for a collision with a ghost |
| 600 | Looks for you eating an energiser. If you do it changes the ghost's colour to 207 (white) and sets a flag 22 moves ahead, when the ghosts will be made to turn back. |
| 610 | Checks for a screen clearance again |
| 620 | Checks if ghosts are almost ready to turn back and beeps if they are |
| 630 | Turns ghosts back |
| 640-660 | Chooses random directions for the ghosts |
| 670-690 | Checks to see if a particular ghost can move in the direction chosen |
| 700 | Adds on ghost's new directions |
| 710-730 | Checks to see if ghost is hitting a wall. If so it stops it, else it POKEs its new position. |
| 740-780 | Memorises what the ghosts are walking over so that it can be POKEd back behind them again. |
| 790 | Checks again (this time numerically instead of PEEKing) to see if we've collided with a ghost |
| 800 | Checks if the ghost was pink or white when we collided with it |
| 810 | Sets up loop |
| 820-860 | End of screen routine |
| 870-930 | End of game routine |

```
360 RESTORE
370 CH=1:PO=1:B=1024:M1=B+176:M2=B+175:M3=B+173:M6=-1:M7=1:M5=1:EE=128:CLSO:PRIN
T@480,"SCORE";:PA=B+303:CC=239:DO=208:SC=0:SD=1
380 ZA=128:ZB=128:ZC=128
390 FORI=0 TO 479:READA
400 IF A=1 THEN POKE I+B,175 ELSE POKEI+B,145
410 IF A=2 THEN POKE I+B,223
420 NEXTI.
430 DATA1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,,,,,,,
,1,,,,,,,,,,,,,,,1,,,,,,,,1,1,2,1,1,,1,1,,1,,1,1,1,,1,1,1,1,,1,1,1,,1,,1,1,,1,1,
2,1
440 DATA1,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,1,1,,1,1,,1,,1,1,1,1,,1,1,1,1,1,1,1,1,,1
,1,1,1,,1,,1,1,,1,1,,,,,1,,,,,,,,,,,,,,,,,,,,,1,,,,,1,1
450 DATA1,1,1,,1,1,1,1,1,1,1,,1,1,1,1,1,1,,1,1,1,1,1,1,1,,1,1,1,1,1,1,1,1,,,,,,,
,,,1,,,,,1,,,,,,,,,,1,1,1,1,1,1,1,1,,1,1,1,1,1,1,1,,1,1,1,1,1,1,,1,1,1,1,1,1,1,,
1,1,1,1
460 DATA1,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,1,1,2,1,1,,1,,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,,1,,1,1,2,1,1,,,,,1,,,,,,,,,,,,,,,,,,,,,1,,,,,1,1,1,,1,1,1,1,1,1,1,,1,
1,1,1,1,1,1,1,1,1,,1,1,1,1,1,1,1,,1,1,1,1,1,,,,,,,,,,,,,,,,,,,,,,,,,,,,,1,1,1,1,1,
1,1,1,1,1,1
470 DATA1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
480 A$=CHR$(128)+CHR$(128):PRINT@173,A$+A$+A$;:PRINT@238,A$+A$;
490 PRINT@301,"READY?";:SCREENO,1:PLAY"T6O2CDEFGAB":FORI=1TO2000:NEXT:PRINT@301,
A$+A$+A$;
500 FORTY=1TO100000:POKEPA-PO,128:IF PEEK(PA)=145 THEN PLAY"O2T255A":SC=SC+10:PR
INT@485,SC;:DO=DO-1
510 IF DO=0 THEN 820
520 POKEPA,159:AD$=INKEY$
530 IF AD$="K" THEN PW=-32
540 IF AD$="M" THEN PW=32
550 IF AD$="X" THEN PW=1 .
560 IF AD$="Z" THEN PW=-1
570 IF PEEK(PA+PW)<>175 THEN PO=PW
580 PA=PA+PO:IF PEEK(PA)=175 THEN PA=PA-PO:PO=0
590 IF PEEK(PA)=207 OR PEEK(PA)=239 THEN 800
600 IF PEEK(PA)=223 THEN PLAY"T255O1B":CC=207:PQ=TY+22:DO=DO-1
```

**Continued on page 47**

```
610 IF DO=0 THEN 820
620 IF TY=PQ-2 THEN PLAY"T25505B"
630 IF TY=PQ THEN CC=239
640  J1=RND(4):IFJ1<3 THEN M9=-3+2*J1 ELSE M9=-96+64*(J1-2)
650  J1=RND(4):IFJ1<3 THEN MA=-3+2*J1 ELSE MA=-96+64*(J1-2)
660 J1=RND(4):IF J1<3 THEN MB=-3+2*J1 ELSE MB=-96+64*(J1-2)
670 IF PEEK(M1+M9)<>175 THEN M5=M9
680 IF PEEK(M2+MA)<>175 THEN M6=MA
690 IF PEEK(M3+MB)<>175 THEN M7=MB
700 M1=M1+M5:M2=M6+M2:M3=M3+M7
710 IF PEEK(M1)=175 THEN M1=M1-M5 ELSE POKE M1-M5,ZB
720 IF PEEK(M2)=175 THEN M2=M2-M6 ELSE POKE M2-M6,ZA
730 IF PEEK(M3)=175  THEN M3=M3-M7 ELSE POKE M3-M7,ZC
740 ZA=PEEK(M2):ZB=PEEK(M1):ZC=PEEK(M3):POKEM2,CC:POKEM1,CC:POKEM3,CC
750 IF ZA=239 OR ZA=159 OR ZA=207  THEN ZA=ZD
760 IF ZB=159 OR ZB=239 OR ZB=207 THEN ZB=ZE
770 IF ZC=159 OR ZC=239 OR ZC=207 THEN ZC=ZF
780 ZD=ZA:ZE=ZB:ZF=ZC
790 IF PA=M1 OR PA=M2 OR PA=M3 THEN 800 ELSE 810
800 IF CC=239 THEN 870 ELSEPLAY"T25502BAGFEDC":SC=SC+100:PRINT@485,SC;
810 NEXTTY
820 RESTORE:FORI=0TO478:READA
830 IF A=1 THEN POKEB+I,255 ELSE POKE B+I,128
840 NEXT            850 RESTORE
860 GOTO390
870 PRINT@299,"GAME OVER!";
880 SCREEN0,1
890 FORI=255T01 STEP-7:SOUNDI,1:NEXT
900 FORI=1TO5000:NEXT
910 CLS:PRINT"YOU SCORED ";SC;"POINTS.";
920 IF INKEY$="" THEN 920
930 GOTO360
```

# Keybeep

*From G D Snooks in Horsham*

THIS SHORT MACHINE code program makes use of the fact that Basic jumps to certain locations during its operation and returns for no apparent reason.

The address the program uses is hex 16A. The location is POKEd with hex 7E to cause it to jump to the Keybeep routine before returning to Basic every time a key is pressed. POKEing 16A with hex 39 returns back to Basic missing the Keybeep routine (see REM lines 100 and 110).

The actual noise is my favourite but can be changed with a little understanding of machine code. This program can interfere with the printer interface and should be turned off before a printer is used, or all you get is END OF LINE CHR$.

```
10 ' KEYBEEP
20 ' ######################
30 ' # G.D.S (C) 18/7/83 #
40 ' ######################
50 CLEAR72,&H7FB8
60 DATA34,36,8E,0,1,10,8E,0,1,30,1,86,E0
,B7,FF,20,7F,1,12,7F,1,13,B6,FF,23,8A,8,
B7,FF,23,B6,FF,1,84,F7,B7,FF,1,B6,FF,3,8
4,F7,B7,FF,3,73,FF,20,34,10,30,1F,26,FC,
35,10,10,BC,1,12,2D,EF,8C,0,70,26,C5,35,36,39
70 FORN=&H7FB9 TO &H7FFF
80 READA$:POKEN,VAL("&H"+A$):NEXT
90 POKE&H16B,&H7F:POKE&H16C,&HB9:POKE&H1
6A,&H7E
100 'ON=POKE&H16A,&H7E
110 'OFF=POKE&H16A,&H39
```

# Learn Morse Code

*From Mike Hibbett in Crawley*

THIS PROGRAM was written for people who know Morse code but cannot understand it when it is sent at speed.

The characters are actually sent at about 12 words per minute, but the space between characters can be varied — from several seconds, to the correct spacing for Morse of 12 wpm.

The characters — which may be letters, numbers, or a mixture of both — are sent in a specified number of groups of five, as in the real test. When the transmission has stopped, the characters sent are printed, allowing the user's success to be determined. There is then the option of trying again or ending the program.

The idea for this program came after studying the Basic "sound" function, allowing a tone of varying length to be outputted.

Thus, for each character a series of numbers representing its Morse code is placed in a data statement. A 1 is a "dit", a 3 is a "dah" and a zero is the indication that the code has finished. In unit time, a "dah" is three "dits", so the number actually removed from the array is put in the sound function, multiplied by a constant to give the correct sound length.

Line 130 is of interest since it involves the use of logical operations to decide the range of the random number E that picks a character from L$. The minus signs are included, of course, because of the unusual result of −1 for a true result. ►

```
1 GOTO3
2 CSAVE"MORSE":SOUND10,1:STOP
3 '****************************
4 '*                          *
5 '*MORSE TUTOR - MIKE HIBBETT*
6 '*      G6COQ  4/08/83      *
7 '*                          *
8 '****************************
10 DATA A,1,3,0,0,0,0,B,3,1,1,1,0,0,C,3,
1,3,1,0,0,D,3,1,1,0,0,0,E,1,0,0,0,0,0,F,
1,1,3,1,0,0,G,3,3,1,0,0,0,H,1,1,1,1,0,0,
I,1,1,0,0,0,0,J,1,3,3,3,0,0,K,3,1,3,0,0,
0,L,1,3,1,1,0,0,M,3,3,0,0,0,0,N,3,1,0,0,
0,0,O,3,3,3,0,0,0,P,1,3,3,1,0,0,Q,3,3,1,
3,0,0
20 DATA R,1,3,1,0,0,0,S,1,1,1,0,0,0,T,3,
0,0,0,0,0,U,1,1,3,0,0,0,V,1,1,1,3,0,0,W,
1,3,3,0,0,0,X,3,1,1,3,0,0,Y,3,1,3,3,0,0,
Z,3,3,1,1,0,0,0,3,3,3,3,3,0,1,1,3,3,3,3,
0,2,1,1,3,3,3,0,3,1,1,1,3,3,0,4,1,1,1,1,
3,0,5,1,1,1,1,1,0,6,3,1,1,1,1,0,7,3,3,1,
1,1,0
30 DATA8,3,3,3,1,1,0,9,3,3,3,3,1,0
40 DIML$(35,6)
50 CLS:PRINT:PRINT"INPUT DELAY BETWEEN C
HARACTERS    BETWEEN 1 AND 100";:INPUT Z
60 PRINT:PRINT:PRINT"PLEASE WAIT......"
70 FOR A=0 TO 35:FOR B=0 TO 6:READ L$(A,
B):NEXT B,A
80 CLS:PRINT:PRINT:INPUT"INPUT TEXT:LETT
ERS ONLY(1),       MIXED(2),NUMBERS ONLY(
3)";A
90 CLS:PRINT:PRINT:INPUT"NUMBER OF 5 CHR
 GROUPS:1 TO 100";B
100 DIMA$(5*B-1)
110 FOR C=0 TO B-1
120 FOR D=0 TO 4
130 E=-(A=1)*(RND(26)-1)-(A=2)*(RND(36)-
1)-(A=3)*(RND(10)+25)
140 A$(5*C+D)=L$(E,0)
150 FOR F=1 TO 6
160 G=VAL(L$(E,F)):IFG=0 THEN 180
170 SOUND 180,1.5*G
180 NEXT F
190 FOR X=0 TO 50*Z: NEXT
200 NEXT D
210 FOR X=0 TO 150*Z:NEXT
220 NEXT C
230 CLS
240 PRINT" THOSE WERE AS FOLLOWS;"
250 FOR H=0 TO B-1
260 FOR I= 0 TO 4
270 PRINT A$(5*H+I);
280 NEXT I
290 PRINT"  ";
300 NEXT H
310 PRINT:PRINT:PRINT"ANOTHER GO?(Y/N)"
320 X$="":X$=INKEY$:IFX$=""THEN320
330 IFX$="Y"THEN 110
340 END
```

# The puzzle maker

*From Michael Jennings in Dodworth*

THIS PROGRAM creates one of those puzzles in which a set of words is hidden in a grid of letters.

When run, the program will first ask for the title of the puzzle and then the number of words. After the words have been typed in, there will be a delay of about three minutes as the words are fitted into the grid. The computer then prints out the puzzle and asks if you would like another copy.

The program was written for use with the Seikosha GP 100A but could easily be converted to use other types of printer.

```
10 '****************************
20 '***** THE PUZZLE MAKER *****
30 '*****  BY M.JENNINGS   *****
40 '*****     26/7/83     *****
50 '****************************
60 CLEAR 5000:CLS:DIM G$(19,19),S$(50)
70 INPUT "TITLE OF PUZZLE";TI$
80 INPUT "HOW MANY WORDS";Z:IF Z>50 THEN PRINT "TOO MANY":GOTO 80 ELSE DIM W$(Z)
90 FOR X=1 TO Z
100 PRINT "WORD NO.";X;:INPUT W$(X):IF LEN(W$(X))>20 THEN PRINT "TOO BIG TRY AGA
IN":GOTO 100 ELSE S$(X)=W$(X):NEXT X
110 CLS0:PRINT@265,"PLEASE WAIT";:SOUND 90,5
120 FOR X=0 TO 19:FOR Y=0 TO 19:G$(Y,X)=" ":NEXT Y,X
130 FOR X=1 TO Z:D=RND(2):IF D=1 THEN NEXT X ELSE T$="":FOR Y=LEN(W$(X)) TO 1 ST
EP -1:T$=T$+MID$(W$(X),Y,1):NEXT Y:W$(X)=T$:NEXT X
140 M=Z
150 F=0:FOR I=1 TO M-1
160 IF LEN(W$(I))>=LEN(W$(I+1)) THEN 180
170 T$=W$(I):W$(I)=W$(I+1):W$(I+1)=T$:F=1
180 NEXT I:IF F=1 THEN M=M-1:GOTO 150
190 FOR X=1 TO Z
200 D=RND(4):ON D GOSUB 490,530,570,610
210 NEXT X
```

**Continued on page 51**

# NEW DRAGON 32 SOFTWARE

**DRAGON CHARSET (C) £7.50**

TEST ON THE HI-RES SCREEN. YES THIS UTILITY LETS YOU WRITE TEXT DIAGONALLY, BACKWARDS, BIG, SMALL PLUS MANY OTHER VARIATIONS. ALTOGETHER AN EXCELLENT PROGRAMMING TOOL.

**DRAGON OTHELLO (C) £5.95**

PROBABLY THE BEST VERSION OF THIS ANCIENT CHINESE GAME EVER TO BE SEEN ON A DRAGON. HOURS OF ENJOYMENT.

DEALER INQUIRIES WELCOME

*SEND CHEQUES/PO TO:*

**ASN COMPUTERS LTD**
**DEPT DU**
**89 STATION CRESCENT**
**ASHFORD, MIDDX TW5 3HN**
**TEL: 0782-57599**

★ *CAN YOU WRITE GOOD QUALITY PROGRAMS? WE PAY TOP ROYALTIES FOR GOOD SOFTWARE. SEND FOR DETAILS*

# DRAGON/ATOM/ORIC-1

**ADD LOW-COST I/O POWER WITH A VIA BOARD**

Drive extra I/O devices, from LEDs and switches to extra peripherals. Link 2 micros for data/program transfer, spool listings to a second micro to print/save etc (see TEVLINK below).

**HARDWARE:** 6522 VIA provides 16 I/O lines + control, serial port, 2 timers, interrupt register. I/O, voltage and interrupt lines taken to 26-way IDC plug for easy connection to peripherals etc. Connectors available for Dragon, Atom. The Board may be interfaced to most other 6502/6809 micros, and may be shared between m/c's by changing the plug-in connector.

**SOFTWARE:** 'TEVLINK' package provides 2-machine parallel link (2 Boards needed), overall speed about 8K DATA bytes/second with error checking. A PRINT SPOOLER is included (only 1 Board needed): connect printer port to VIA board on receiving micro and use normal LIST/PRINT etc.

Dragon/Atom cassette .......... **£7.75**
Assembled board + Atom connector .......... **£18.75**
Assembled board + Dragon connector .......... **£18.75**
Dragon 14K EPROM Board: takes 3x4K and 1x2K EPROMS. Specify 25 or 27 series. Assembled Board .......... **£15.00**

**PROFESSIONAL QUALITY DRAGON SOFTWARE:**
TML-DATABASE: A flexible, general-purpose data filing and retrieval program, with an "Auto-Reminder" feature for birthdays, appointments etc, plus a powerful search facility. M/code program .......... **£9.75**

**ORIC ADVENTURE:** Tevrog's Kingdom is a real-time text adventure for the 48K Oric. It features spells, clues and problems, as well as enemies and allies — can you steal King Tevrog's Talisman before he returns? Although no two games are the same, the distribution of items around the layout is logical rather than totally random, giving a game which retains its interest even after you have succeeded several times. Excellent value for money. Cassette ..... **£5.75**

★ *Other software and hardware available soon* ★

**MAIL ORDER ONLY, SEND FOR LISTS/DETAILS**
**All prices fully inclusive of VAT, p&p etc**

**Tevward Microtech Ltd**
**(Dept DU), 403 Dallow Road, Luton LU1 1UL**
**Telephone: (0582) 418906**

DU



```
220 M=Z
230 F=0:FOR I=1 TO M-1
240 IF S$(I)<=S$(I+1) THEN 260
250 T$=S$(I):S$(I)=S$(I+1):S$(I+1)=T$:F=1
260 NEXT I:IF F=1 THEN M=M-1:GOTO230
270 FOR R=0 TO 19:FOR C=0 TO 19:IF G$(R,C)=" " THEN G$(R,C)=CHR$(RND(26)+64)
280 NEXT C,R
290 FOR R=0 TO 19
300 FOR C=0 TO 18
310 G$(R,C)=G$(R,C)+"  "
320 NEXT C,R
330 CLS:PRINT " PREPARE PRINTER":SOUND 150,50
340 PRINT #-2,CHR$(14):PRINT #-2,STRING$(5," ")+TI$:PRINT #-2,CHR$(15)
350 T=INT(Z/5):IF T<>Z/5 THEN T=T+1
360 FOR X=1 TO T
370 PRINT #-2,S$(X),S$(X+T),S$(X+T*2),S$(X+T*3),S$(X+T*4)
380 NEXT X
390 PRINT#-2:PRINT#-2
400 FOR R=0 TO 19
410 R$="":PRINT #-2
420 FOR C=0 TO 19:R$=R$+G$(R,C):NEXT C
430 PRINT #-2,STRING$(10," ")+R$
440 NEXT R
450 CLS:INPUT "ANOTHER COPY (Y/N) ";A$
460 IF A$="Y" THEN :PRINT#-2,CHR$(13):PRINT #-2,CHR$(13):GOTO 340
470 IF A$="N" THEN END
480 GOTO 450
490 R=RND(20)-1:C=RND(20-LEN(W$(X)))-1
500 CC=C:FOR Y=1 TO LEN(W$(X)):IF MID$(W$(X),Y,1)=" " THEN NEXT Y ELSE IF G$(R,C
C)<>" " AND G$(R,CC)<>MID$(W$(X),Y,1) THEN 490 ELSE CC=CC+1:NEXT Y
510 FOR Y=1 TO LEN(W$(X)):G$(R,C)=MID$(W$(X),Y,1):C=C+1:NEXT Y
520 RETURN
530 R=RND(20-LEN(W$(X)))-1:C=RND(20)-1
540 RR=R:FOR Y=1 TO LEN(W$(X)):IF MID$(W$(X),Y,1)=" " THEN NEXT Y ELSE IF G$(RR,
C)<>" " AND G$(RR,C)<>MID$(W$(X),Y,1) THEN 530 ELSE RR=RR+1:NEXT Y
550 FOR Y=1 TO LEN(W$(X)):G$(R,C)=MID$(W$(X),Y,1):R=R+1:NEXT Y
560 RETURN
570 R=RND(20-LEN(W$(X)))-1:C=RND(20-LEN(W$(X)))-1
580 RR=R:CC=C:FOR Y=1 TO LEN(W$(X)):IF MID$(W$(X),Y,1)=" " THEN NEXT Y ELSE IF G
$(RR,CC)<>" " AND G$(RR,CC)<>MID$(W$(X),Y,1) THEN 570 ELSE RR=RR+1:CC=CC+1:NEXT
Y
590 FOR Y=1 TO LEN(W$(X)):IF MID$(W$(X),Y,1)=" " THEN NEXT Y ELSE G$(R,C)=MID$(W
$(X),Y,1):R=R+1:C=C+1:NEXT Y
600 RETURN
610 R=20-RND(20-LEN(W$(X)))-1:C=20-RND(20-LEN(W$(X)))-1
620 RR=R:CC=C:FOR Y=1 TO LEN(W$(X)):IF MID$(W$(X),Y,1)=" " THEN NEXT Y ELSE IF G
$(RR,CC)<>" " AND G$(RR,CC)<>MID$(W$(X),Y,1) THEN 610 ELSE RR=RR-1:CC=CC-1:NEXT
Y
630 FOR Y=1 TO LEN(W$(X)):IF MID$(W$(X),Y,1)=" " THEN NEXT Y ELSE G$(R,C)=MID$(W
$(X),Y,1):R=R-1:C=C-1:NEXT Y
640 RETURN
```

# Slow down your listing speed

*From J Whitmore in Leicester*

"CONGRATULATIONS, YOU have saved the earth, your final score was −13.753" — clearly my score was being tampered with. Not for the first time did I regret the Dragon's supercharged LIST which forced either listing short blocks or attempting continual SHIFT @.

My own interest in a solution to this problem stemmed from an article in the American magazine *Rainbow* in February which offered a joystick controlled variable speed lister for the Tandy Colour Computer, a machine not unlike the Dragon. Sadly this machine code patch utilised several ROM calls which were machine dependent . . . I returned to SHIFT @.

But a recent article in *Popular Computing Weekly* revived my interest. In it reference was made to location 359 which Basic visits as part of the 'print to screen' routine. I recalled that this was one of the major requirements restricting my previous progress. With the location entered and minor amendments to the joystick addresses it was complete. The finished routine controls the speed of listing via the right joystick and as it creates a delay before printing characters to the screen should also improve the trace facility.

```
10 REM *** VARILIST ***
20 CLEAR 200,31999
30 FOR X=32000 TO 32024:READ Y:POKE X,Y:NEXT
40 DATA 52.86,189,189.82,190,1,91,48
50 DATA 1,48,31,18,18,19,38,249,53,86
60 DATA 57,38,249,53.86,57
70 POKE 359,126:POKE 360,125:POKE 361,0
80 NEW
```

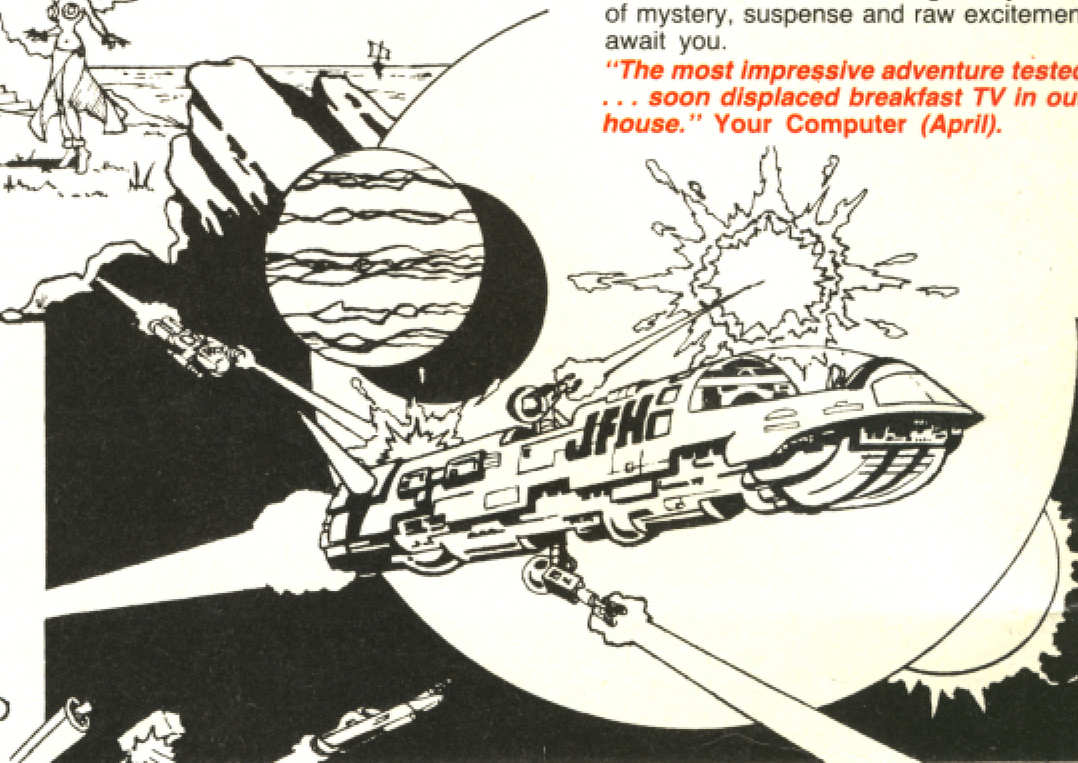
# DRAGON 32



## THE RING OF DARKNESS

The game created to be the most complete, enthralling, addictive and detailed adventure ever written for the Dragon 32. A whole graphic world (of nearly 3,000 travel days) with kings, princesses, mad jesters and dangerous quests will unfold before you. Across ocean, river, mountain and forest you travel to trade in many towns and dare the depths of vast dungeons (more than nine levels deep with over 20 different creatures in hi-res 3-D graphics), in search of fortune and the hidden curse of the Dark Ring. Many hours of mystery, suspense and raw excitement await you.

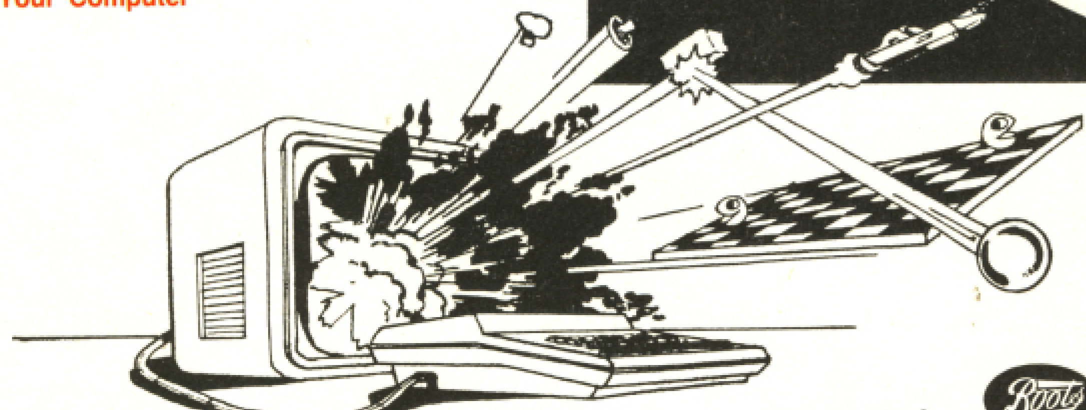
***"The most impressive adventure tested . . . soon displaced breakfast TV in our house."*** **Your Computer (April).**



## DRAGON TREK

Your starship has just entered quadrant 6-8. You are surrounded by eight battle cruisers of the Klingon Empire. Do you fight or run? They launch a salvo of photon torpedoes. You're HIT! Get a damage report. It's OK, the shields are holding, the next move is up to you, but be quick or they will attack again. Hi-res graphics, onboard computer and lag time interaction will have you zapping Klingons until stardate 2477.

***"Excellent value for money."***
**Your Computer**



## PEPPER'S GAME PACK

Huge blue mushrooms, mad wizards, bouncing bricks, chessboard strategy, doomed space shuttles and mind teasing codes. They're all to be found on this great compendium of six games (Mushroom Muncher, Nad the Necromancer, Pepper's Breakout, Chessboard, Tracker, Chameleon Run and MasterBrain.)

**Available from Boots and good software outlets**

## ARTIST'S DESIGNER

Produce intricate designs in hi-res graphics and save them on cassette. Full list of features includes text merging, joystick or keyboard control, circles, paint and shape replication. Ideal for home video captions, education, charts and diagrams.

**All prices include p&p, VAT and the WINTERSOFT guarantee of quality and reliability**

WINTERSOFT SOFTWARE

*Dealer enquiries welcome*

**Tel: 01-367 5720**

Post coupon now to:

**WINTERSOFT**
**Dept. D, 30 Uplands Park Road, Enfield, Middx EN2 7PT**

Please rush me copies of (tick box)

- ☐ The Ring of Darkness .......... £10.00
- ☐ Dragon Trek .......... £6.99
- ☐ Pepper's Game Pack .......... £7.95
- ☐ Artist's Designer .......... £6.99

I enclose cheque/PO for £ ..........

NAME ..........

ADDRESS ..........

..........

..........

This is the chance to air your views — send your tips, compliments and complaints to Letters Page, *Dragon User*, 12-13 Little Newport Street, London WC2R 3LD.

# Letters

## Merging made easy

MERGING PROGRAMS can be done more easily than in the method show in the August issue of *Dragon User*. It can be done in a few simple steps.

1. LOAD "A"
2. POKE 25,PEEK(27)
3. POKE 26,PEEK(28)-2
4. LOAD "B"
5. RENUM so that "B" starts after program "A"
6. POKE 25,30 : POKE 26,1

You must renumber program "B" so that it starts after program "A", otherwise the two programmes will just tag on to each other. This results in a crash when RUN.

I also have a tip for saving a screen of text on to cassette. This could be useful for loading information at the beginning of a cassette or program.

1. Put information on screen
2. CSAVEM "NAME" ,1204,1535,1024

To load it back, you simply use CLOADM "NAME"

Finally, to disable the keyboard you type POKE 65301,0. To enable it again you type POKE 65301,20. This could be used to disable the keyboard in a program when calculations are being made and to enable it again when an input is required.

*Adrian Hadayah,*
*London E18.*

## Graphics on tape

AS SOME graphics can easily take over 15 minutes to be drawn, I thought it would be much handier to be able to save them on tape so that they can be replaced into memory in a fraction of the time.

My method is based upon the CSAVEM command described in the manual. The graphic pages for the Dragon 32 are at the following locations:

| | |
|---|---|
| Page 1 | hex address 600-BFF |
| Page 2 | C00-11FF |
| Page 3 | 1200-17FF |
| Page 4 | 1800-1DFF |
| Page 5 | 1E00-23FF |
| Page 6 | 2400-29FF |
| Page 7 | 2A00-2FFF |
| Page 8 | 3000-35FF |

PMODE 0 uses only one page. PMODEs 1 and 2 use two pages, and PMODEs 3 and 4 use four pages. So, assuming that your start-page was 1, you can save your graphics using one of the commands below:

CSAVEM "NAME",&H600, &HBFF, &H600 (for PMODE 0)
CSAVEM "NAME",&H600, &H11FF, &H600 (for PMODE 1 or 2)
CSAVEM "NAME",&H600, &H1DFF, &H600 (for PMODE 3 or 4)

Your graphics masterpiece is now on tape. To access it at any time use the CLOADM command CLOADM "NAME" or simply CLOADM.

Once loading is complete, type in the following to display the graphics:

```
10 PMODE 4,1 : SCREEN 1, 1
20 GOTO 20
```

Note that line 10 sets up the graphics mode as it was in your original program. But changing of the PMODE value can give some interesting results.

*D G Rendle,*
*Southampton.*

## Defending the Z80

I ENTIRELY agree with the view expressed in your August issue that the 6809 is nicer to program than the Z80, but I feel that some remarks about the Z80 are unduly severe and misleading to readers.

A rigorous comparison of the two chips would be a considerable task involving many days of work and I have not carried this out. However, I have written thousands of lines of source code for both machines and made some comparisons between them and I would like to point out some saving graces of the Z80.

In the matter of the instruction sets it is universally accepted that the 6809 is easily to be preferred overall. However, the Z80 has the DJNZ (decrement loop counter) and jump if non zero) and LDIR (block move) instructions which one really misses on the 6809. One also misses the conditional CALL and RETURN instructions.

It is also accepted that the 1MHz 6809E has much more power than the 2MHz Z80. But most Z80s are 4MHz Z80As, which may be more powerful than the 6809E. Z80 Bs (6MHz) are available more cheaply than a 6809E and the Z80 H (8MHz) has also been announced, so to say that the 6809 is "emphatically the best 8-bit chip" is contentious.

There is, moreover, one area in which the Z80 beats the 6809 hands down — the Z80 is one of a family of cheap, available VLSI chips which support a very powerful interrupt mode. There is no equivalent to interrupt mode 2 on the 6809, no second set of registers for interrrupt handling and no equivalent for the Z80 A DMA chip, for example, on the British market.

There must be many applications where a Z80-based system would be the logical choice, despite the fact that the 6809 is undoubtedly "nicer to use for programming".

*James W. Thomas,*
*London NW4.*

## Colourful Dragon

WE ALL know that the Dragon has nine available colours. What may not be well known is that it also contains a wide selection of striped paint. If you don't believe it, try the following program:

```
10 PMODE 3,1: SCREEN 1,0
20 FOR I=1 TO 255
30 POKE &HB2,I
40 POKE &HB3, I-1
50 PCLS : CIRCLE (128,96), 75
60 PAINT (128,96)
70 NEXT I
```

It would appear that the standard Basic commands for selecting colours only set the two least significant bits of locations B2 (current foreground colour) and B3 (current background colour), whereas POKEing can set these to any value between 0 and 255. Quite where the stripes come from though, I have no idea.

*K A Blackwell*
*Southampton.*

## Underused function

ONE FUNCTION which seems to be vastly underused — judging by a study of published programs for the Dragon over the past year — is the ASC function.

Its use is quite simple. ASC(X$) will give the ASCII code (American Standard Code for Information Interchange) of the first character of X$. So, if X$="A", then the value returned is 65 (see Appendix A in the manual).

The implications of ASC are not so simple. I have seen several programs recently where the use of ASC would reduce the coding quite considerably. For example:

```
200 IF A$="A" THEN F=15
210 IF A$="B" THEN F=25
220 IF A$="C" THEN F=35
```

and so on down to

```
300 IF A$="K" THEN F=115
```

In this routine there were 11 lines and the pattern is very clear. As A$ is tested in turn for the letters A to K, F is set to the appropriate value from the list 15, 25, 35 . . . 115 — numbers increasing in 10s.

Now the ASCII codes for A, B, C . . . K are 65, 66, 67 . . . 75. So, while the ASCII codes increase by 1, the F values increase in 10s. This pattern can now be translated into a one-line statement to replace the original 11.

```
200 F=(ASC(A$)-64)10+5
```

The section ASC(A$) – 64 reduced the character A$ to a number in the range 1, 2, 3 . . . 11 according to whether A$ is A, B, C . . . K. If this value is now multiplied by 10, then this will produce numbers from the list 10, 20, 30 . . . 110. These values are clearly five too small and so a five must be added.

In looking at this new one-line version, I would agree that the meaning is not as obvious as the earlier 11 lines, but surely that is taking clarity too far! Just think of the time you have saved yourself and the many readers who have tried out your program.

*Paul Yeandle,*
*Poole.*

## Software Top 10

| | | | |
|---|---|---|---|
| 1 | (1) | **Mountain** | Dragon Data |
| 2 | (5) | **The King** | Microdeal |
| 3 | (—) | **Frogger** | Microdeal |
| 4 | (7) | **Android Attack** | Microdeal |
| 5 | (5) | **Space War** | Microdeal |
| 6 | (8) | **Nightflight** | Salamander |
| 7 | (—) | **Shuttle** | Microdeal |
| 8 | (9) | **Planet Invasion** | Microdeal |
| 9 | (—) | **Ring of Darkness** | Wintersoft |
| 10 | (—) | **Cuthbert Goes Walkabout** | Microdeal |

Chart compiled by Boots

If you've got a technical question or problem write to Cathy Hyde, Dragon Answers, *Dragon User*, 12/13 Little Newport Street, London WC2R 3LD.

# Dragon Answers

## Saving trouble

I HAVE a problem saving programs on my computer.

The recorder is taping something but the stored information, when loaded, is greeted by an I/O error. I believe this is due to the level of the sound omitted by the computer when saving — it is far too loud.

Is there anything that can be done about it?

*A. Hughes,*
*Swansea.*

**YOU DO not state whether your cassette recorder has automatic recording level. This may sometimes cause the program to be saved at a level too low for the Dragon 32.**

**However, you say that the sound emitted by the Dragon is "too loud" which may indicate that you have some form of interference. You could try moving your television back from the Dragon, and it may help to remove your mic lead when loading and ear lead when saving.**

**It is also worthwhile trying all different volume settings and checking that the heads are clean.**

**If problems persist and you feel the fault could lie with the machine or leads, you could return it to the original dealer for testing.**

## Machine code

PLEASE COULD you tell me where to get hold of the machine code booklet mentioned in your magazine?

Also, is there any way that the audio output can be taken from the Dragon direct, without it going through the UHF modulator to the TV? It would then allow the audio to feed into a better quality amplification system.

*M Elliott,*
*Brough,*
*E Yorkshire.*

**THE AUDIO output may be obtained at the monitor socket of the Dragon. The sound is available on pin 1 of the monitor socket which is the first pin going clockwise as you look at the socket on the Dragon (see diagram).**



**All requests for the *Information for machine code users* booklet should be directed to Dragon Data enclosing a SAE. The company's address is Kenfig Industrial Estate, Margam, Port Talbot, Wales.**

## Recorder suggestion

I HAVE had a Dragon 32 computer since Christmas and have been unable to find a tape recorder that will work with my computer.

Boots recommended a Bush recorder and I tried two, but they would not load or save programs. I have also tried Ferguson and Sharp recorders, neither of these were able to both load and save.

I would be pleased if you could recommend a tape recorder that will work with my Dragon 32.

*Mathew Hendey,*
*Ryde,*
*Isle of Wight.*

**THE PROBLEM with recommending a specific recorder is that manufacturers may change their sources from time to time. Although a recorder might still appear in the same casing, under the same model number, there might be slight differences such as head setting, giving problems with loading and saving.**

**I don't know if this is the case with your Bush 3150 but I have had success using this recorder myself, although on some lengthy programs it has proved helpful to remove the EAR lead when saving and MIC lead when loading.**

**The problems with the Ferguson 3T07 can usually be alleviated by fitting a 470K Ω resistor in series with the MIC lead of the cassette interface lead — this is easiest at the DIN end of the lead.**

**Other advice is to check heads are clean, try all volume and tone levels, and to keep the television slightly away from the Dragon.**

## Equation problems

COULD YOU please help me with the equation for getting inverse sine and inverse cosine? The one given in my manual for inverse cos is (ATN(X) SQR (−X*X+1)) + 1.5708. How does the Dragon evaluate ATN(X)SQR? — mine doesn't! The argument for inverse sine has the computer trying to find the square root of a minus number, which is Nobel prize stuff.

Please can you help me and give me some idea of what to use?

*Tony Pratt,*
*Avon.*

**THE INVERSE cos given in the manual is not correct and is unfortunately due to a misprint. I suggest you try the following for function of inverse cosine of X and function of inverse sine of X:**

**(i) DEF FNC(X) = π/2 − ATN(X/SQR(1−X ↑ 2))**

**(ii) DEF FNS(X) = ATN(X/SQR(1−X ↑ 2))**

## Places to POKE

CAN YOU please tell me why when I type the following program into my Dragon, the screen fills up with another program and will not accept Basic commands such as NEW?

```
10 CLS 0
20 FOR I = 1 TO 31 STEP 4
30 POKE 1, ASCCCHR$(159))
40 NEXT I
```

Will this damage my Dragon?

*Karen Beaumont,*
*Huddersfield.*

**THE REASON you are getting garbage with this program is that you are POKEing into the direct page which is extensively used by Basic. This will cause many problems such as those you have encountered.**

**To avoid this you shouldn't POKE anywhere below &H400 (buffers and direct page) or above &HIE00 (generally Basic and variable storage).**

**The only definite "safe" place to POKE to is above the CLEAR statement.**

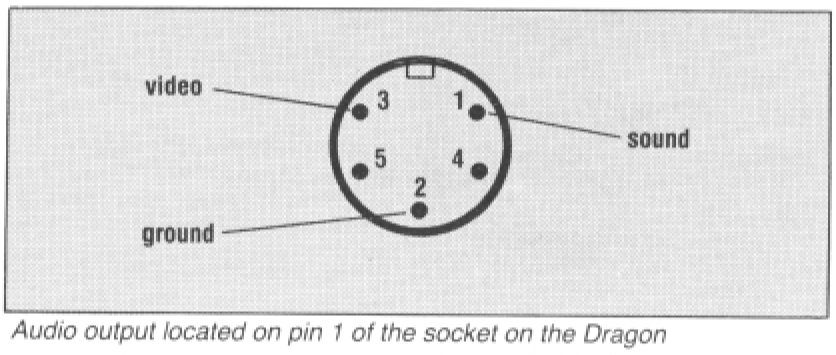
## Frozen pictures

I HAVE a cartridge, Meteoroids, which I play a lot. However, after a couple of minutes playing the picture freezes. Is there any explanation for this?

*David Jones,*
*Cheddleton,*
*Staffs.*

**CARTRIDGES SHOULD not normally "freeze" during use and this may indicate a problem with the connector or the machine itself.**

**It is difficult to tell exactly where the problem lies so I suggest you return your machine, with the cartridge, to the original dealer or your nearest service agent, for testing.**



*Audio output located on pin 1 of the socket on the Dragon*

# Classified

**ANYONE FOR CRICKET?**
Choose the number of overs.
Set your fielders to suit your bowler.
10 bowling options. Full scoreboard.
Save game facility.
1 Joystick required.
**£4.95 inc p&p**
**DORSOFT, 15 GURJUN CLOSE, UPTON, POOLE, DORSET BH16 5QH.**

**DISASSEMBLER MONITOR CARTRIDGE**
Fed up with BASIC —
Then use MACHINE CODE
Over 20 Powerful Commands
'DISMON' for the Dragon 32
Send **£18.86** to:
**KayJay Electronic Services**
**13 The Boulters, Gorleston**
**Great Yarmouth NR31 6TF**

**AERIAL LEADS,** 2 metres £1.80, 3 metres £1.99. Dragon-type double joystick extensions, 2 metres £6. Dragon listings £2, tape returned, aerial splitter combiner £2.20. Further details SAE to: M. Swanson, High Street, Spilsby, Lincs.

**DRAGON USERS.** Send your program on tape, will return with listing, £2. N. W. Phillips, 190 Frimley Green Road, Frimley Green, Camberley, Surrey.

**DRAGON 32K,** joysticks, printer cable, programs, manuals, leads, as new, £174. Tel: Atherton 884738.

**DRAGON HARDWARE**
**Input Output Module.** 4 or 8 Inputs. 1 to 8 Outputs
Input Rating - 2 to - 15 volts. Output Rating 220V AC 0.5 Amp.
Price: 4 Input **£19.82**. 8 Input **£27.82**.
Relay Ouputs Extra **£3 per Output**
**Memory Module.** CMOS non-Volatile Memory. Stores and retains programs when Dragon is switched off.
Price: 4K Bytes **£42.22**.
8K Bytes **£50.22**. 14K Bytes **£62.22**.
Modules plug into cartridge port users instructions provided. P&P **£1.50** per module.
Large SAE for further details.
**MORAY MICROSYSTEMS**
**17 KIRKTON PLACE, ELGIN**
**MORAY, SCOTLAND IV30 3JR**

**COMPUTER COVERUP.** Attractive dust cover for Dragon, £2.95, including postage; Printer etc covers, price on request. Computacover, 52 Wolseley Road, Rugeley, Staffordshire.

**DRAGON WRITER.** Mix text and Hi-res Graphics, different sizes, colours, angles. A sub-routine to begin all your programs. £3.50. Samyh Software, 9 Earlswells Drive, Cults, Aberdeen.

**DRAGON DELTA DISK OWNERS.** Catalogue your hi-fi record collection then you can locate any record title artist or track title in your collection. Functions include: input, amend, enquire, browse, print of record details. Menu driven. Colour display. Send cheque for £7.95 payable to: Lincsroxy Ltd, 20 Plovers Mead, Wyatts Green, Doddinghurst, Brentwood, Essex.

**LISTING SERVICE FOR DRAGONS.** Send your program on tape plus large SAE and £1 for two lists per program. Same day despatch to: Mr Ryder, 28 Barnard Close, Vange, Basildon, Essex SSI 64SH.

**EDUCATED DRAGON DIET STATISTICS**
17+K very friendly program full data editing — cassette storage and graphical presentation. **£4.95**
**Educational Games in full colour and sound for 4+ year old. Edusoft 1 covers alphabet, numbers and counting. Edusoft 2, 3 Games of basic Maths.**
Visually entertaining and musical **£5.95** each inclusive.
**GOLD SOFTWARE, 6 REDDITCH ROAD, HOPWOOD, WORCS. B48 7TL.**

**COMPUTER CORRESPONDENCE COURSES**

A complete course in BASIC programming which is designed specifically for the DRAGON micro-computer. It includes word processing, file-management, stock-control, games, and educational applications. Suitable for the complete beginner, hobbyist, teacher, and business person.

Cost of Course
**£89.50** (incl)
Price per Module: **£8.95** (incl)
*For full details write enclosing sae to*
**COMPUTER CORRESPONDENCE COURSES**
**(Dept. D.U.) Churchway, West Ilsley Nr Newbury BERKS RG16 0AH**
**Tel: (06328) 336**

**PYRAMID ADVENTURE** for Dragon 32. Travel in search of the Pharaoh's lost treasure but beware of the curse. Send £4.95 for cassette to R. Baugh, 36 Penrice Drive, Tividale, Warley, West Midlands B69 1UQ.

**DRAGON LISTINGS** from Cassette by return of post, each program £1.25, extra copies 50p. M. J. Lees, 38 Wem Road, Shawbury, Salop.

**★★ STOCKMARKET ★★**
Enter the Dragon world of high finance with this new game for the Dragon 32. Buy and sell shares on the stock exchange
— An exciting and addictive game —
Available only from
**N. Van Braddr**
**72 Berry Hill Lane**
**Mansfield, Notts.**
**£4.95**

**DRAGON PHOTOGRAPHERS.** Store your negatives on data tapes, search by subject or description, record exposure details, etc, up to 200 negatives per file, £4.95 inc p&p. J.M. Photography & Engineering, 9 Abbey Lane, Sedgebrook, Grantham, Lincs.

**BCS Ltd**
**Non-Dragon Data hardware and software**
£1.50 brings your first copy of our constantly updated list of other suppliers of goodies specifically for your Dragon. (£1 refund or 2nd year free, for **new** subscribers full details of any supplier not already on our lists.)
**BCS Ltd, 47A Queen Street**
**Newton Abbot, Devon TQ12 2AU**

**DATA DUPLICATION** for the Dragon, from 38.1p. Tel: 08692 2831 anytime.

**PROTECT YOUR MACHINE CODE PROGRAMS.** Make your own spare copy. For instructions send £1 + sae to Dragoncopy, 63 Riversdene, Stokesley, Middlesbrough, Cleveland, TS9 5DD.

**FROGHOP:** Inglewood Software's first in a series of hi-speed computavoice programs. Joysticks and computavoice optional, hi-res machine code action for only £3.95 inclusive. Inglewood Software, 18 Magheralave Road, Lisburn, Northern Ireland BT28 3 BN.

**PROGRAM PRINTING SERVICE.** Hard copy from your cassettes. Up to 10K, £2. Free quotation on receipt of sae. Send to Willow Software, Willow Bank, Ilston, Swansea SA2 7LD.

**TRY THIS!!!**
Type 'POKE 65495,1'. If your computer returns 'OK' and a fast flashing cursor then RESET and . . .
**BUY THIS!!!**
**★ THE SOFT SEVEN ★**
A collection of seven quality programs for the DRAGON 32
1. GALACTIC BOMBER
2. HEAR AND SPELL
3. WORDSEARCH
4. GRAPHIX
5. MUSIC MAKER
6. HANGMAN
7. THE GREAT RACE
All on one cassette costing **ONLY £3.50** incl P&P. Send cheque postal order to:
**T. D. COPSEY**
**20 Thorington Avenue, Daws Heath Hadleigh, Benfleet, Essex**
*★★★★★FAST SERVICE★★★★★*

**MACE EDITOR ASSEMBLER MONITOR**
Cartridge with 80-page manual. This well documented product provides an easy way to m c programming, for both professional and hobbiest. **£29.95. M C PROGRAMMERS PACK** contains the data sheets for the Dragon's four main ICs (6809 MPU, 6821 PIA, 6847 VDG, 6883 SAM). Also includes 6809 reference card. **£3.95.** 6821 PIA IC. **£1.95. DRAGBUG M C MONITOR & DISSASSEMBLER** cassette. **£12.95. FLIGHT SIMULATOR £7.95!** SAE brings details of games, business, educational and utility programs.
**ELTEK, 29 Parkside Avenue, Littlehampton, Sussex, BN17 6BG.**

**GETTING INTO MACHINE CODE?**
**WE HAVE THE PASSWORD**
Which lets you into the heart of the DRAGON ROM.
PASSWORD is much more than a mere disassembler. It interprets and executes each instruction in sequence, giving a reliable line by line listing of the routines you want to use. Multi-screen operation for user-friendliness, with or without a printer. Use it also to create and debug your own software.
**For £6 from:**
**B. FRANKLIN, 4 CHURCH CIRCLE, FARNBOROUGH, HANTS.**

**GREAT DRAWING,** why not print it out with our Dragon screen dump? Yes, only £7.95 by return post, works on your Acorn Seikosha GP100A, Tandy DMP100. No quibble guarantee. Order from Caveman Computers Co Ltd, 55 Iona Road, Windy Nook, Gateshead, NE10.

**HARDWARE ★ for the ★ DRAGON 32**
Analog Input Board — features — Fast (32$\mu$S) Conversion 8 Bits — 0-5V or 0-10V Input — Price (with application notes): £29.50.
Relay Output Unit
Control up to eight relays. Contacts rated at 5A at 250V AC. Runs from printer socket.
Price with two relays fitted **£16.75.**
Extra two relays (Max. eight) add **£6.50**
*Send SAE for further details*
**PNP COMMUNICATIONS**
**62 LAWES AVENUE, NEWHAVEN EAST SUSSEX BN9 9SB**

**THE SWORD IN THE STONE.** Can you save the princess before your dragon devours her? An adventure game for the Dragon 32. Send £4.95 to Modo Productions, 3 Victoria Street, Ainsworth, Bolton, BL2 5RB.

**METRIC CONVERSIONS,** giving areas, volumes and capacities of cubes, tubes, etc. Also Central Heating Calculator, heat losses, radiator sizes, boiler size, etc, for up to 14 rooms, £7 each. Torwood Educational Tapes, 250 Leicester Road, Markfield, Leicester.

**DRAGON OWNERS** — 2 New Unique Utility programs for your Dragon. 1. Basic Aid — speed program entry by single key input of common Basic commands. Redefine any or all keys. Merge, move and renumber any part of your program. Cartridge $49.95. 2. The Stripper — Deletes remarks, packs lines and removes blanks. The Stripper takes off all that excess fat in your program. With the Stripper your programs will run faster and take up much less memory. Cassette $19.95. Order both and save $10. All prices in US dollars and include post packing and insurance. Spectrum Projects, 93-15 86th Drive, Woodhaven, NY 11421 USA.

**ADVENTURE TEXT** and high resolution graphics, £3. Send to 29 Firthwood Road, Coal Aston, Sheffield S18 6BW.

**DRAGON 32.** RS232C compatible printer interface, connects your Dragon 32 printer port to any RS232 compatible printer, just plug it in and you're ready to go. Ideal for word processing programs, ie, telewriter converts to 7 bits for ASCII data or all 8 bits for pure binary data. Runs at 300 baud rate. Comes complete with V24 25-pin female connector for your printer, plus 20-pin parallel centronics connector which plugs directly into your Dragon printer port. Price £30 inc p&p. Obtainable from: Comtec, 7 Grove Meadow, Welwyn Garden City, Herts AL7 2BE. Tel: Welwyn Garden City 34596.

**DRAGON 32 LISTINGS** from your program on cassette (cassette returned with listing), £1.25. Send SAE and cassette to P.G. Computer Services, 6 Azalea Avenue, Gorton, Manchester M18 8PQ.

**SMALL BUSINESS DRAGON SOFTWARE.** Phone list, address list, stock list, order list, cash flow. £5.99 each cassette, £23.99 five cassettes. Alexander Systems, 45 Shackleton Spring, Stevenage, Herts.

ELKAN ELECTRONICS

**Don't let your Dragon breathe fire! Don't fry your chips!**

**DRAGON'S EYE**

On-off indicator for Dragon 32/ Tandy Color machines. 10-second installation — no soldering required.

**Only £2.95 each (p&p free)**

**ELKAN ELECTRONICS**, FREEPOST. 11 Bury New Road, Prestwich, Manchester M25 6LZ Tel: 061-798 7613

**DRAGON WORD PROCESSOR.** Cassette based. Full features, 20K text buffer, full user manual included. £10, from R. J. Crowther, 141 Hollyfield, Harlow, Essex.

**ED AND EXPLODE!** Two challenging games for your Dragon. £7.95, or send £1 and any Dragon tape or cartridge and receive this new cassette in exchange! The Rock Shop, Tideswell, Derbyshire.

**ACCURATE ASTROLOGY FROM ASTROCALC,** £18; Natal longitudes, declinations, aspects, equal, placidus, Koch Houses, etc, £33, plus Progressions, Transits. Large SAE: 67 Peascroft Road, Hemel Hempstead HP3 8ER. Tel: 0442 51809.



# Here's my classified ad.

**(Please write your copy in capital letters on the lines below.)**

| | | | |
|---|---|---|---|
| £0.20 | £0.40 | £0.60 | £0.80 |
| £1.00 | £1.20 | £1.40 | £1.60 |
| £1.80 | £2.00 | £2.20 | £2.40 |
| £2.60 | £2.80 | £3.00 | £3.20 |
| £3.40 | £3.60 | £3.80 | £4.00 |
| £4.20 | £4.40 | Please continue on a separate sheet of paper | |

I make this................................ words, at 20p per word so I owe you £................................

**Name**..........................................................

**Address**..........................................................

..........................................................

..........................................................

**Telephone**..........................................................

Please cut out and send this form to: Classified Department, *Dragon User*, 12-13 Little Newport Street, London WC2R 3LD

# Competition Corner

**Answers to Competition Corner, *Dragon User,* 12/13 Little Newport Street, London WC2R 3LD**

# Link your Dragon to the outside world

*Gordon Lee gives you the chance to win a modem and interface*

OF ALL THE "classes" into which numbers can be put, that of the prime numbers must surely be one of the most interesting, intriguing, mystifying, unusual, unpredictable, and quite likely, downright infuriating of them all.

In fact, prime numbers are the very basic "building blocks" of our number system. All numbers are either prime or composite (non-prime), and every composite number can be broken down into a unique series of prime factors. For example, 156 has the prime factors 13, 3, 2 and 2.

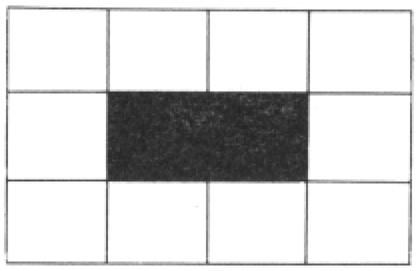
Despite its simple definition — a prime number is a number with no factors other than itself and 1 — there remain many mysteries regarding primes. For centuries mathematicians have endeavoured to find a formula which would generate only prime numbers — and failed!

However, we can state confidently that there are an infinite number of primes.

Not quite so certain is the theory of Christian Goldbach, who stated that every even number is the sum of two primes. For example, 36 is the sum of 31 and 5. Since 1742, when this conjecture was first announced, no one has succeeded in coming up with a proof that this definitely is the case — but, on the other hand, no one has been able to discover a single instance where this theorem does not apply.

Below is a simple cross-number grid, in which I would like you to fit eight primes! I say *eight,* because each number can be read in both directions — from left to right and from right to left in the case of the horizontal numbers, and from top to bottom and bottom to top in the case of the vertical numbers.

So, for example, if you used the prime 197 you would find that its reverse, 791, was *not* prime so this figure would not do. Also, you are only allowed to use each digit — zero to nine — once and only once. **Noting that mere reversals, reflections or rotations of the same grid *do not* count, how many *different* solutions can you find?**






## Prizes

THE WINNER OF this month's prize will have access to a wider world of communications thanks to Cotswold Computers and CP Engineering. The two companies are offering an RS232 interface plus modem, enabling the winner to call up anybody else using a modem that meets British Standard specifications.

Cotswold — which has opened its own Dragon specialist shop at 6 Middle Row, Chipping Norton, Oxfordshire — is confident that the demand for communications facilities will grow, with an RS232 interface being an essential first step in linking your Dragon to the outside world. While Cotswold markets the interface, it was designed by CP Engineering, as a result of work it did on data logging applications.

The acoustic coupler modem is originate-only, which means it can call up any other modem that is operating on the same frequency. CP Engineering is also working on an originate and answer modem which Cotswold will market. Other projects include an 8K back-up Ram device which will prevent Dragon users losing what they've stored when they switch off. CP Engineering is on Malvern 68008 and Cotswold on (0608) 41232.

## Rules

TO WIN THE modem plus RS232 interface, you have to send in the most elegant solution to the puzzle. You must show both the answer to the competition and how to solve it with the use of a Basic program developed on your Dragon 32. As a tie-breaker, complete the following sentence in 15 words or less:

I want to add a modem to my Dragon 32 in order to ..............................................

..............................................................

..............................................................

..............................................................

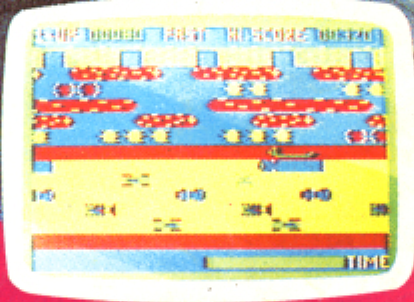
Your entry must arrive at *Dragon User* by the last working day in October 1983. The name of the winner, and the solution to the puzzle, will be printed in the December issue of *Dragon User*. You may only enter the competition once. Entries will not be acknowledged and we cannot enter into correspondence on the final result. Please send your entries to *Dragon User*, 12/13 Little Newport Street, London WC2R 3LD.

## August winner

THE ANSWER TO the August competition is that, apart from base 10, the digits 225 represent a perfect square in the following bases: 61, 358 and 2089. These represent the decimal equivalents of 7569 ($87^2$), 257049 ($507^2$) and 8732025 ($2955^2$) respectively. The winner is Mrs J C Wheeler of Hayling Island in Hampshire who will be receiving her prize of a colour monitor from Statacom and Portatel.






TIME